Étude de l'Urbanisation des Accès Virtuels et Stratégie de Métamorphose de Réseaux

Oussama Stiti

THESE DE DOCTORAT DE
L'UNIVERSITE PIERRE ET MARIE CURIE

Spécialité

Informatique
Ecole doctorale Informatique, Télécommunications et Electronique (Paris)

Présentée par

Mr. Oussama Stiti

Pour obtenir le grade de

DOCTEUR de l'UNIVERSITÉ PIERRE ET MARIE CURIE

Sujet de la thèse :

Étude de l'Urbanisation des Accès Virtuels et Stratégie de Métamorphose de Réseaux

soutenue le 15 décembre 2015
devant le jury composé de :

Dr. Nadjib Achir	Rapporteur, HDR, Université Paris 13. Épinay-sur-Seine, France.
Dr. Lila Boukhatem	Rapporteur, HDR, Université Paris-Sud. Orsay, France.
Pr. Otto Carlos Duarte	Examinateur, UFRJ. Rio De Janeiro, Brésil.
Pr. Pascal Urien	Examinateur, Télécom Paristech. Paris, France.
Dr. Anne Fladenmuller	Examinateur, HDR, UPMC (Paris6). Paris, France.
Dr. Othmen Braham	Co-encadrant de thèse, VirtuOR. Paris, France.
Pr. Guy Pujolle	Directeur de thèse, UPMC (Paris6). Paris, France.

Remerciements

Au terme de mes travaux, je tiens à exprimer toute ma gratitude aux personnes qui ont contribué de près ou de loin au bon déroulement de cette thèse. Tout d'abord, Je tiens à sincèrement remercier Pr. Guy Pujolle de m'avoir accueilli au sein de l'équipe PHARE au LIP6. Il a su m'apporter un encadrement sans failles et des conseils avisés dont j'ai puisé l'inspiration pour le bon déroulement de mes travaux de recherche.

Ensuite je tiens à remercier chaleureusement Dr. Othmen Braham de m'avoir accueilli au sein de l'entreprise VirtuOR. Il a su m'apporter un support technique et une rigueur scientifique indéniables pour l'avancement de mes travaux de développement de thèse.

J'adresse également mes remerciements à l'entreprise VirtuOR ainsi qu'à tous ses membres pour m'avoir offert un environnement adéquat à l'accomplissement de mes travaux.

Je remercie tous mes collègues de l'équipe PHARE qui m'ont accompagné durant ces trois années dans la joie et la bonne humeur.

Je tiens à remercier les examinateurs du jury, Pascal Urien, Anne Fladenmuller, Otto Carlos Duarte et tout particulièrement les rapporteurs Lila Boukhatem et Nadjib Achir de m'avoir accordé de leur temps et de leur expertise.

Merci à tous mes amis pour leur support et encouragements, j'ai puisé en eux ma motivation et mon acharnement durant cette thèse. Merci spécialement à Gorgi, Sarah, Yara, Ghorbel, Hssine et Chaaben.

Et finalement je remercie tout particulièrement mes parents Hassouna Stiti et Wahida Stiti ainsi que ma sœur Sabrine sans qui je n'en serais pas là aujourd'hui, ils m'ont apporté un soutien sans failles durant toutes ces années. Ils ont contribué à mon développement personnel et intellectuel. Ils m'apportent quotidiennement de la joie de vivre, et l'envie de les rendre fiers. A vous, je vous dédis cette thèse.

Résumé

La virtualisation, originalement introduite dans les réseaux pour en réduire les coûts de maintenance et de déploiement, a connu une explosion fulgurante remodelant le paysage des réseaux informatiques et télécoms. La virtualisation permet la mutualisation des ressources physiques pour instancier des machines virtuelles complétement isolées mais puisant leurs ressources du même matériel physique. Plus récemment le NFV (Network Functions Virtualisation) est apparu, et a permis de virtualiser des classes entières de fonctions de nœud de réseau dans des blocs qui peuvent se connecter pour créer des services de communication. Cette thèse s'inscrit dans ce contexte pour virtualiser les nœuds des réseaux d'accès, à savoir les points d'accès Wi-Fi. Le Wi-Fi est devenu la technologie de tous les enjeux pour les opérateurs mobile. Cette technologie leur permet notamment d'y délester une partie du trafic de données clients via des hotspots. Le problème qui se pose dans un tel mécanisme est que les normes Wi-Fi existantes ainsi que les logiciels de gestion de connexion d'un appareil mobile n'ont pas été développés dans l'optique du hotspot. A cet effet, la norme Hotspot2.0 a été créée, pour rendre l'expérience Wi-Fi similaire à celle du cellulaire en termes d'itinérance, de transparence et de sécurité. Nous avons dans nos travaux, appliqué le concept de NFV en virtualisant ces points d'accès Wi-Fi de nouvelle génération. La problématique face à laquelle nous avons été confrontés est la forte sécurité imposée par de tels dispositifs exigeants notamment l'enregistrement et l'installation de certificats de sécurité clients dans les lieux publics. Dans notre thèse nous proposons une architecture innovante permettant le rapatriement de ces éléments de sécurité à travers des bornes NFC. Ces mêmes bornes, dans une volonté d'urbanisation des points d'accès, permettront aux utilisateurs de créer leurs propres points d'accès Wi-Fi virtuels à la volée. Enfin, le dernier aspect de cette thèse touche à la problématique de gérance des entités virtualisées changeant les schémas de communication des réseaux traditionnels. Dans ce contexte, SDN (Software Defined Network) a émergé dans les datacenters pour redéfinir la façon de penser les réseaux plus en adéquation avec le contexte virtualisé. Cette thèse reprend le SDN pour l'appliquer en périphérie de réseaux sur les points d'accès Wi-Fi virtuels que nous avons créés. Plus qu'un nouveau paradigme de

communications réseaux, nous verrons que l'introduction des concepts NFV/SDN aux réseaux Wi-Fi permettra dans un avenir proche de rendre les réseaux Wi-Fi plus souples, plus ouverts et plus évolutifs.

Mots Clés : Virtualisation, Wi-Fi, NFV, Hotspot2.0, NFC, SDN.

Abstract

Virtualization was originally introduced in networks to reduce maintenance and deployment costs. It has experienced tremendous growth and reshaped the landscape of IT and ITC networks. Virtualization permits the sharing of physical resources for instantiating isolated virtual machines despite the fact that it is drawing its resources from the same physical hardware. More recently NFV (Network Functions Virtualization) appeared, it allows to virtualize entire classes of network functions node in blocks that can connect to create communication services. In this thesis we virtualize the access network nodes, namely Wi-Fi access points. The Wi-Fi became one of the hot-topic technology for mobile operators, it allows them to offload some of the customers' data traffic via hotspots. The problem that arises in such a mechanism is the existing wireless standards, and mobile devices connection management software have not been developed for this purpose. The Hotspot2.0 standard was created to overcome this limitation, by making the Wi-Fi experience similar to the cellular in terms of roaming, transparency and security. We have in our work, applied the concept of NFV by virtualizing these brand new Wi-Fi access points. One of the problems that we faced is the high security required by such standard, including the provisioning of client credentials in public areas. In our thesis we propose an innovative architecture for the repatriation of these credentials through NFC terminals. These same terminals will be used for access points' urbanization by allowing users to create their own Wi-Fi virtual access point on the fly. The last aspect of this thesis is related to the management of virtualized entities changing communication patterns of legacy networks. In this context, SDN (Software Defined Network) emerged in data centers to redefine the way we think about networks, and is designed for virtualized environments. In this thesis we brought SDN to the edge of the network in our Wi-Fi virtual access points. More than a new paradigm of networks communications, we will see that NFV/SDN in Wi-Fi networks will in the near future make Wi-Fi networks more flexible, open and scalable.

Keywords: Virtualization, Wi-Fi, NFV, Hotspot2.0, NFC, SDN.

Table des matières

Introduction Générale

De nos jours, la raison principale qui pousse les industries IT à virtualiser la plupart - ou la totalité - de leurs infrastructures informatique est d'ordre financière. La virtualisation aide à réduire drastiquement les coûts. Prenons par exemple le cas d'une infrastructure de serveurs tournant tous à 30% de leur capacité. Nous avons tout intérêt à augmenter la capacité d'utilisation de ces serveurs pour éliminer les serveurs physiques de trop engendrant des coûts d'achat, de fonctionnement et de maintenance supplémentaires. Un serveur sur six serait inutilisé dans le monde, pour un coût global de plus de 25 milliards de dollars par an [1]. De par sa définition, la virtualisation permet une démarcation nette entre le matériel et le logiciel. La virtualisation permet la mutualisation des ressources physiques pour instancier des machines virtuelles tout à fait isolées mais puisant du même matériel à savoir la mémoire, le CPU et les interfaces réseaux nécessaires à leur bon fonctionnement. Grâce à la virtualisation, il est possible d'exécuter différentes instances de nœuds de réseau (routeurs, points d'accès, PABX etc.) sur une même ressource physique. L'allocation des ressources dans un environnement virtuel a connu une évolution rapide et plusieurs outils commencent à être fournis à cet effet. Les instances virtuelles peuvent remplir de manière identique les fonctionnalités qu'offrent les équipements réseaux (routeur, point d'accès, firewall, etc.), les équipements utilisateurs (poste de travail, poste à domicile, etc.) et les équipements serveurs (serveur Mail, serveur WEB, etc.).

En parallèle de ce changement profond que connaissent les réseaux à travers la virtualisation, une autre tendance est en plein essor : le client est de plus en plus mobile. L'ubiquité des réseaux Wi-Fi combinée à l'intégration low-cost des chipsets Wi-Fi dans tous les appareils mobiles font du Wi-Fi la technologie sans fil la plus utilisée pour accéder à Internet [2]. Le trafic de données mobile connait une croissance à une vitesse vertigineuse. En 2014, on comptait 2.5 exabyte (10^{18} byte) de données mobiles échangées par mois [3]. Le nombre des ventes de tablettes et smartphones a augmenté rapidement et ces terminaux nécessitent un accès Internet quasi-permanent. Les clients ont besoin d'être

connecté partout et à tout moment. Les applications aussi sont devenues de plus en plus gourmandes en bande passante, à cause des contenus multimédias de plus en plus volumineux.

Les hotspots Wi-Fi ont, dans un sens, toujours été en compétition avec les réseaux de données cellulaires. Aujourd'hui ce n'est plus le cas. En effet, le Wi-Fi est en train de devenir une partie intégrante des stratégies des opérateurs de données mobiles dans le monde. Les fournisseurs utilisent le Wi-Fi, qui est relativement peu couteux à déployer, pour délester (offloading) des données cellulaires et améliorer la qualité de service [4]. La Wi-Fi Alliance s'est penchée sur la nécessité de 'décharger' les réseaux cellulaires vers les réseaux Wi-Fi, en créant la norme Hotspot2.0. Celle-ci permet de passer de façon transparente d'un réseau cellulaire à un réseau Wi-Fi (et vice-versa). Nous étudierons cette norme en profondeur dans cette thèse. Nous avons développé dans nos travaux une version virtualisée de ces nouveaux points d'accès Wi-Fi respectant cette norme.

C'est dans ce contexte de virtualisation et de mobilité que nous nous positionnons dans cette thèse pour répondre aux besoins d'accès des clients dans un contexte d'opérateurs. La mutualisation des ressources des bornes d'accès Wi-Fi, la possibilité d'instancier une multitude de points d'accès Wi-Fi virtuels intrinsèquement différents les uns des autres sur une même ressource physique, et la conceptualisation du Wi-Fi logiciel constituent les enjeux que nous évoquerons dans cette thèse. Le développement de points d'accès Wi-Fi virtuels nous permet d'envisager plusieurs scénarios possibles quant à leurs utilisations. Il ne sera pas étonnant à l'avenir que plusieurs opérateurs se partagent les mêmes points d'accès physiques pour y instancier leurs points d'accès Wi-Fi virtuels. On peut aussi envisager des opérateurs allouant les ressources de leurs bornes Wi-Fi à un opérateur concurrent pour pouvoir y instancier un point d'accès Wi-Fi virtuel. C'est cette mutualisation d'infrastructure d'accès que nous appelons urbanisation des points d'accès. C'est ce concept qui va nous servir de fil conducteur tout au long de cette thèse.

L'omniprésence des points d'accès Wi-Fi dans les lieux publics tels que les aéroports, les hôtels, les restaurants ou plus communément dans la rue offre aux pirates informatiques une multitude de portes d'entrée pour effectuer des actions malveillantes. Les brèches d'entrée sont nombreuses, elles peuvent se situer au niveau de la borne Wi-Fi en elle-même, au niveau du client ou de façon

plus répandue entre le point d'accès Wi-Fi et le client (man in the middle). La connexion à un point d'accès Wi-Fi dans un lieu public se fait souvent sans chiffrement, permettant à n'importe qui sur ce réseau de récupérer toutes les sessions, mots de passes et informations confidentielles des clients. De façon générale et même avec la présence de chiffrement à l'aide d'un mot de passe, la connexion à des réseaux Wi-Fi publics est déconseillée si l'usage dont on veut en faire concerne l'accès à un compte ou des données ayant de la valeur. Le Hotspot2.0 permet entre autre d'atteindre un niveau de sécurité dans les réseaux Wi-Fi comparable à celui des réseaux cellulaires [5]. Pour ce faire, la Wi-Fi Alliance recommande l'utilisation d'authentification à base de certificats de sécurité (niveau de sécurité utilisée dans les réseaux Wi-Fi d'entreprise) mais sans spécifier comment distribuer ces certificats dans un environnement Wi-Fi public. Pour pallier à ce problème, nous avons proposé dans notre thèse une architecture permettant de rapatrier de façon sécurisée les certificats nécessaires à un client pour se connecter à un point d'accès Wi-Fi virtuel public (cette proposition reste valable si le point d'accès ne supporte pas la virtualisation) grâce à des bornes NFC. Ces bornes peu coûteuses placées dans des lieux publics permettront aux clients désirant se connecter de façon sécurisée à un point d'accès, de rapatrier un certificat de sécurité.

SDN peut avoir un pouvoir de transformation sur les réseaux de forte densité. À ce jour, les discussions autour de SDN ont largement été axées sur leur utilisation dans les datacenters. Quand les entreprises sont passées à la virtualisation et au cloud, ils ont constaté que la configuration des réseaux au sein des data center est fastidieuse et est sujette aux erreurs. SDN est apparu pour remédier à ce problème en permettant une certaine auto-configuration du réseau. C'est sur ce même schéma qu'on se base pour apporter SDN en périphérie du réseau étant donné l'environnement virtualisé que nous avons créé sur le réseau d'accès. Nous verrons entre autre dans cette thèse comment les points d'accès Wi-Fi OpenFlow que nous avons développés permettent un accès plus souple dans des réseaux Wi-Fi denses et hétérogènes.

Contexte

Cette thèse a débuté en janvier 2013, dans le cadre d'une collaboration entre le LIP6 et l'entreprise VirtuOR. Il s'agit d'une thèse CIFRE alliant des sujets de recherche à des problématiques industrielles.

VirtuOR est une entreprise qui a vu le jour en 2008 à Paris. Elle est le fruit d'une collaboration entre l'université du Québec au Canada et l'université Pierre et Marie Curie à Paris. L'entreprise commercialise des équipements de réseaux virtuels (routeur virtuel IPv4 ou IPv6, Label Switch Router, points d'accès Wi-Fi virtuels, serveur SIP virtuel, PBX virtuel, etc.). La technologie VirtuOR permet le déploiement et le déplacement à la volée de multiples instances de machines virtuelles sur les équipements physiques du réseau tout en offrant des propriétés d'étanchéité entre chaque instance virtuelle.

Concrètement la solution VirtuOR permet à une petite ou moyenne entreprise de mettre en place plusieurs réseaux virtuels, spécifiques aux applications dont elle a besoin. VirtuOR commercialise, avec ses équipements physiques, un logiciel d'urbanisation capable d'instancier à la volée des machines virtuelles selon le besoin de l'utilisateur en un simple clic.

Durant des années, VirtuOR s'est efforcé de développer les concepts d'urbanisation des réseaux sans fils ainsi que les *metamorphosing networks* [6] : dans le but de faciliter l'accès d'un utilisateur à ses services avec le minimum d'intervention de sa part, le réseau d'accès doit se métamorphoser selon le besoin du client. Ces recherches mettent en relief les contraintes croissantes pour qu'un utilisateur se connecte convenablement à ses services informatiques indépendamment de son environnement en lui proposant des points d'accès innovant. Ces points d'accès innovants intègrent des apports introduits par VirtuOR pour la création d'un environnement de réseaux virtuels. Ils peuvent ainsi se métamorphoser plus facilement selon le contexte de l'utilisateur.

L'objectif de cette thèse est d'améliorer la solution des points d'accès Wi-Fi virtuels VirtuOR déjà existante, et de prendre ce point de départ pour proposer des nouvelles technologies et architectures permettant de densifier le portfolio des points d'accès Wi-Fi proposés par VirtuOR. De plus, au court de nos travaux, nous avons constaté que l'environnement sans fil se complexifiait de plus en plus du fait de son hétérogénéité. A cet effet, une architecture réseau permettant de

gérer et de visualiser l'ensemble des nœuds de notre réseau fût introduite à travers le SDN.

Structure du rapport

Ce manuscrit de thèse s'articule autour de trois chapitres. Etant donné le fait que les technologies évoquées recouvrent un large spectre, pour simplifier la structure de ce manuscrit, nous donnerons un état de l'art en début de chaque chapitre. Le chapitre 1 présente de manière précise la virtualisation sous tous ses aspects. Nous y évoquons notamment l'enjeu de la virtualisation des réseaux, mais aussi les technologies sous-jacentes qui permettent la coexistence de machines virtuelles hétérogènes sur le même substrat physique. Nous expliquerons ensuite dans ce même chapitre le concept de point d'accès Wi-Fi virtuels, leur fonctionnement et la solution que nous avons développée durant cette thèse.

Le chapitre 2 s'intéresse à la gestion de la mobilité des clients dans un environnement virtualisé, et s'axe en deux parties. Dans la première partie, nous étudierons un des nouveaux paradigmes des réseaux Wi-Fi : La norme Hotspot2.0 ainsi que de la virtualisation de son architecture réseau que nous avons réalisée. Dans la deuxième partie de ce chapitre nous étudierons une architecture réseau se basant sur des bornes NFC que nous avons conçue pour permettre le rapatriement de certificats de sécurité et/ou la création de points d'accès Wi-Fi virtuels dans un contexte de bornes Wi-Fi virtualisées dans des zones publiques.

Le chapitre 3 quant à lui présente les réseaux reposant sur le SDN, et toutes leurs importances dans les réseaux sans fils. Nous y expliquons en outre tout l'intérêt et l'enjeu de cette dernière contribution dans l'environnement sans fil virtuel développé tout au long de cette thèse. Pour ce faire, nous y présenterons les points d'accès Wi-Fi OpenFlow que nous avons développés ainsi que le contrôleur SDN qui permet d'introduire une entité de contrôle pour la supervision des réseaux sans fils.

Chapitre 1 : La virtualisation et son application aux points d'accès Wi-Fi

1.1 Introduction

Plus que jamais, dans presque toutes les industries, les sociétés ont été amenées à moderniser leurs réseaux informatiques en y rajoutant une composante de virtualisation. Un des points clés de ces nouveaux réseaux concerne la démarcation entre le matériel et les systèmes d'exploitation. La virtualisation dans le cœur du réseau permet à la fois l'isolation entre les instances virtuelles, et le partage des ressources physiques entre différents domaines de gestion. Grâce à la virtualisation, il est possible d'exécuter différentes instances de nœuds de réseau sur une même ressource physique. La virtualisation peut être considérée comme faisant partie d'une tendance générale dans l'entreprise IT pour intégrer les systèmes autonomiques : un scénario dans lequel l'environnement informatique sera en mesure de se gérer lui-même.

La virtualisation a été adoptée plus rapidement que ce que les experts avaient imaginé. Il y a trois domaines de l'informatique où la virtualisation s'est déployé en masse : la virtualisation de réseau, la virtualisation du stockage et la virtualisation des serveurs:

- La virtualisation du stockage est la mise en commun du stockage physique à partir de plusieurs périphériques de stockage réseau dans ce qui semble être un dispositif de stockage unique qui est géré depuis une console centrale. La virtualisation du stockage est couramment utilisée dans les réseaux de stockage (Storage Area Networks: SANs).
- La virtualisation des serveurs consiste à masquer les ressources du serveur (y compris le nombre et l'identité des serveurs physiques, les processeurs et systèmes d'exploitation) aux utilisateurs finaux. L'intention est d'épargner à l'utilisateur d'avoir à comprendre et gérer les détails complexes des ressources du serveur tout en augmentant la capacité et l'utilisation de ces ressources.

- La virtualisation de réseau est une méthode pour combiner les ressources disponibles dans un réseau en divisant la bande passante disponible dans les canaux, dont chacun est indépendant des autres et peut être affecté (ou réaffecté) à un serveur ou dispositif particulier dans un slot de temps précis. L'idée est que la virtualisation déguise la véritable complexité du réseau en le séparant en parties gérables en dépassant la contrainte d'un matériel dédié. La virtualisation de réseau est plus communément appelée NFV (Network Functions Virtualisation), nous l'aborderons plus en détail dans la sous-section 1.2.2.

C'est sur ce dernier aspect de la virtualisation que nous nous focaliserons en partant du principe que l'on peut virtualiser tout type de nœud réseau. C'est dans cette démarche que nous nous inscrivons dans notre thèse pour virtualiser les équipements en bordure de réseau à savoir les points d'accès Wi-Fi. Nous verrons ainsi dans ce chapitre les fondements de base de la virtualisation, ainsi que les outils que nous avons choisis pour permettre une telle mise en pratique. C'est ensuite que nous aborderons le vif de notre sujet en présentant la virtualisation des points d'accès Wi-Fi, avec tout l'environnement, la technique, et les enjeux qu'ils représentent.

1.2 La virtualisation

1.2.1 Définition

La virtualisation consiste à séparer le matériel physique du logiciel en émulant le matériel à l'aide de logiciels. Quand un système d'exploitation différent fonctionne au-dessus du système d'exploitation primaire au moyen de virtualisation, il est désigné en tant que machine virtuelle. Une machine virtuelle n'est rien d'autre qu'un fichier de données sur une machine physique qui peut être déplacé et copié sur une autre machine physique, de la même façon qu'un fichier de données normal. Les machines virtuelles ainsi créées fonctionnent à l'identique des machines physiques, et sont isolées les unes des autres malgré l'utilisation du même substrat physique. Pour parvenir à séparer la couche physique de la couche logicielle, il faut utiliser une couche d'abstraction supplémentaire qu'on appelle

hyperviseur. Un hyperviseur ou moniteur de machine virtuelle (Virtual Machine Monitor : VMM) est un logiciel qui crée et exécute les machines virtuelles. C'est ainsi l'hyperviseur qui permettra d'instancier plusieurs machines virtuelles fonctionnant chacune avec son propre système d'exploitation mais partageant les mêmes ressources physiques présentes sur la couche matérielle (CPU, cartes réseaux, RAM, etc.).

Schématiquement, on représente la virtualisation comme une architecture tripartite (Cf. Figure 1) où la couche d'hypervision joue le rôle d'intermédiaire entre la couche physique et les machine virtuelles. Chaque machine virtuelle a son propre système d'exploitation et sa propre couche applicative en fonction du besoin qu'on désire en faire. Il est à noter que le terme « Hardware virtuel » dans la Figure 1 désigne les ressources misent à disposition de la machine virtuelle à partir de la couche physique via l'hyperviseur. Ce hardware virtuel est configurable lors de la création d'une machine virtuelle en mentionnant à l'hyperviseur les ressources physiques dont la machine virtuelle a besoin pour fonctionner.

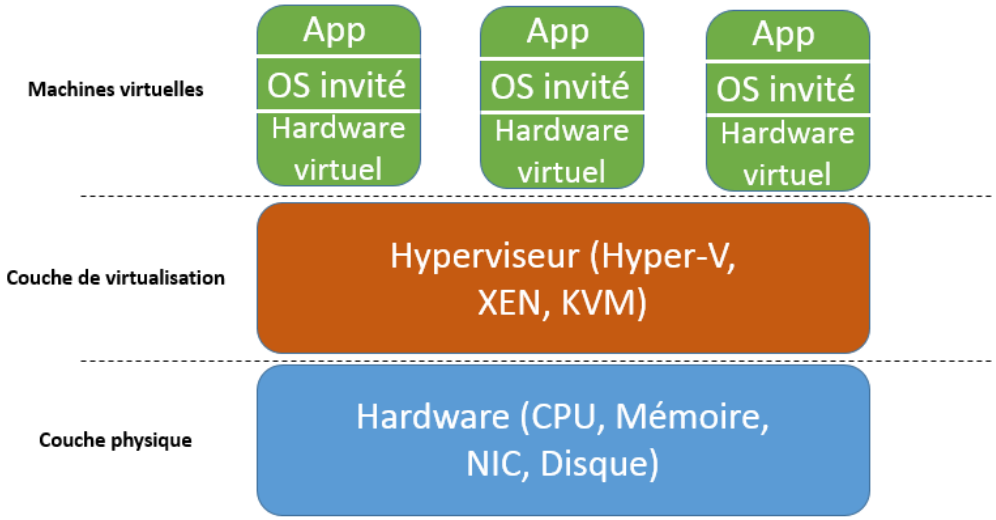

Figure 1: Vue globale de la virtualisation

1.2.2 NFV

La technologie NFV a été abordée pour la première fois en octobre 2012 par le groupe 'Network Function Virtualisation' de l'ETSI (European

Telecommunications Standards Institute) dans un white paper [7]. Le NFV est un concept d'architecture de réseau qui utilise les technologies de virtualisation afin de virtualiser des classes entières de fonctions de nœud de réseau dans des blocs qui peuvent se connecter pour créer des services de communication. Une fonction de réseau virtualisée (Virtual Network Function : VNF), est constituée d'une ou de plusieurs machines virtuelles en cours d'exécution avec des logiciels et processus différents, au-dessus de serveurs, switchs et unités de stockage de grandes capacités

Le but de NFV est de découpler les fonctions réseau du matériel physique qui y est dédié comme l'illustre la Figure 2. En outre, le NFV permet aux services de réseau qui sont contenus dans les routeurs, pare-feu, équilibreurs de charge et autres périphériques dédiés, à être hébergés sur des machines virtuelles (VM). Une fois que les fonctions réseau sont sous le contrôle d'un hyperviseur, les services qui nécessitaient auparavant un matériel dédié peuvent maintenant être lancés sur des serveurs x86 standards. Cette caractéristique est importante car elle signifie que les administrateurs réseau n'auront plus besoin d'acheter du matériel dédié afin de construire une topologie réseau complète. La capacité d'un serveur sera capable d'être agrandie de façon logicielle, évitant ainsi le surdimensionnement matériel et permettant de réduire les dépenses en capital (CAPex) et en exploitation (OPex). Si une application fonctionnant sur une machine virtuelle nécessite plus de bande passante, par exemple, l'administrateur peut déplacer la machine virtuelle vers un autre serveur physique ou instancier une autre machine virtuelle sur le serveur d'origine pour prendre une partie de la charge. Avoir cette souplesse permettra à l'industrie IT de répondre d'une manière plus souple à l'évolution des objectifs de l'entreprise et les demandes de services réseau.

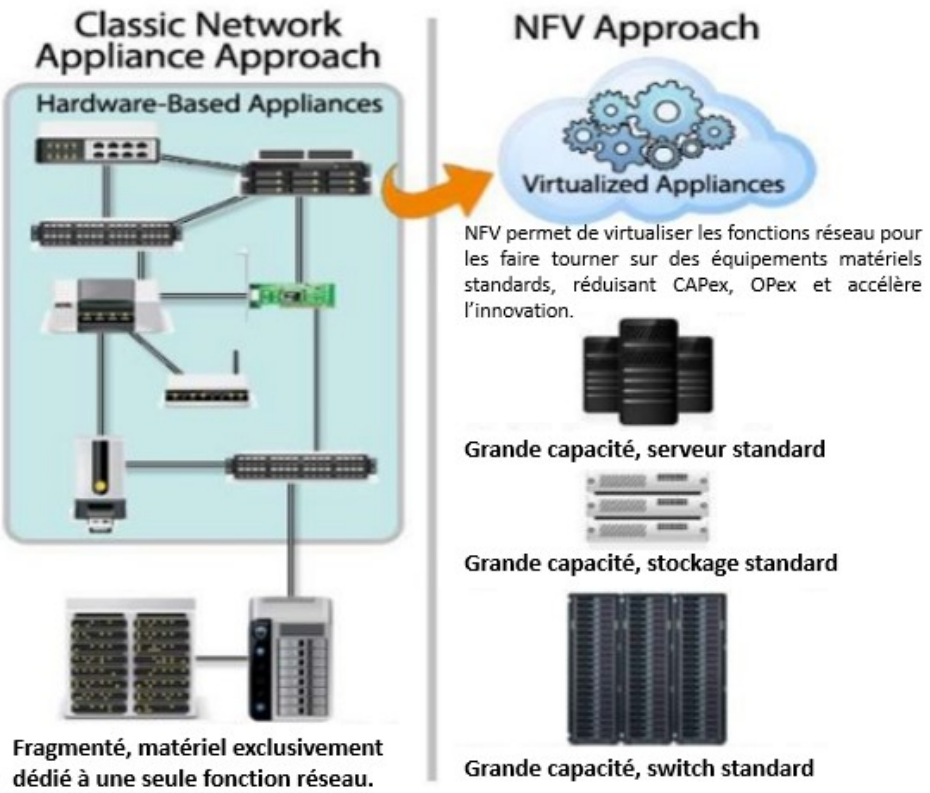

Figure 2: L'approche réseau classique et l'approche NFV

1.3 L'hyperviseur Xen

1.3.1 Définition et historique

On appelle hyperviseur une plate-forme de virtualisation qui permet à plusieurs systèmes d'exploitation de travailler sur une même machine physique simultanément. Xen est un hyperviseur basé sur le concept de micronoyau (micro-kernel), fournissant des services qui permettent à plusieurs systèmes d'exploitation d'être exécutés sur le même matériel informatique en même temps. Historiquement, l'Université de Cambridge a développé la première version de Xen en 2003. Malgré son franc succès, le logiciel a été maintenu en open-source, sous réserve des exigences de la GNU General Public License (GPL) version 2. Xen est responsable de la gestion de la mémoire et de l'ordonnancement du CPU de toutes les machines virtuelles (qu'on appelle des « DomU»). Xen est aussi responsable du lancement du domaine le plus privilégiée : le « Dom0 » (la seule machine virtuelle qui par défaut dispose d'un accès direct au matériel).

L'hyperviseur peut être géré depuis le Dom0, c'est aussi à partir de ce domaine qu'on peut lancer les domaines non privilégiés (DomU : machine virtuelle).

1.3.2 L'architecture de Xen

Comme nous l'avons mentionné ci-dessus, une instance de machine virtuelle est appelée domaine ou DomU. Un domaine spécial, appelé domaine 0 (Dom0) contient les pilotes pour tous les périphériques du système. Le Dom0 contient également une couche de contrôle pour gérer la création, la destruction et la configuration des machines virtuelles.

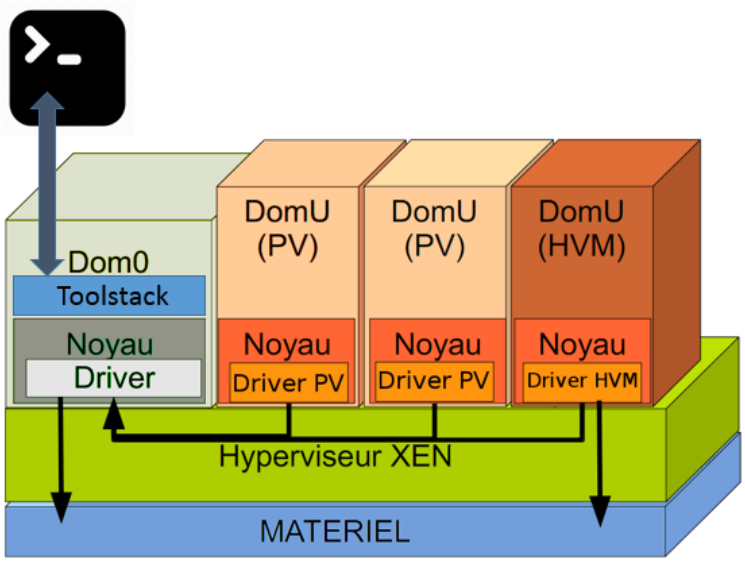

Figure 3: L'architecture de Xen

L'architecture de Xen (Figure 3) est composée de plusieurs couches communiquant les unes avec les autres. Nous pouvons distinguer :

- La couche matérielle, c'est là que sont présents les I/O (input/output), la mémoire, le CPU, les interfaces réseaux.
- L'hyperviseur Xen, logiciel qui fonctionne directement sur le matériel et est responsable de la gestion du CPU, de la mémoire et des interruptions. C'est le premier programme à se lancer après la sortie du *bootloading*.
- Les DomU ou machines virtuelles, sont des environnements virtualisés, chacun exécutant leur propre système d'exploitation et applications.

L'hyperviseur prend en charge deux modes de virtualisation différents: la para-virtualisation (PV) et la virtualisation complète (HVM). Les deux types de modes peuvent être utilisés en même temps sur le même hyperviseur. Les DomU sont totalement isolés du matériel: en d'autres termes, ils n'ont pas de privilège d'accéder directement au matériel ou aux fonctionnalités d'I/O.

- Le Dom0 ou domaine de contrôle, est une machine virtuelle qui possède des privilèges spéciaux tels que la possibilité d'accéder directement au matériel, de gérer tous les accès aux fonctions d'I/O du système et d'interagir avec les autres machines virtuelles. Elle propose aussi une interface de commande, à travers laquelle le système est commandé. L'hyperviseur Xen n'est pas utilisable sans Domain 0, c'est la première VM lancée par le système.
- Le Toolstack, est une couche de contrôle présente dans le Dom0 qui permet à un utilisateur de gérer la création, la destruction et la configuration des machines virtuelles. Le Toolstack présente une interface qui est soit pilotée par une console de ligne de commandes, par une interface graphique ou par des outils d'orchestration du cloud comme OpenStack ou CloudStack.

1.3.3 La virtualisation dans Xen

Nous avons vu précédemment qu'il existe deux types de virtualisation, la virtualisation complète appelée HVM et la virtualisation partielle ou para-virtualisation PV. Lors de l'utilisation de la HVM, le DomU n'a pas conscience d'être virtualisé car l'hyperviseur lui émule un environnement matériel complet. En contraste, avec la PV, le DomU est conscient de s'exécuter dans un environnement virtualisé, et interagit directement avec l'hyperviseur pour transmettre les appels systèmes au matériel de la machine physique hôte.

Il est à noter que la virtualisation complète HVM engendre des performances souvent médiocres dues à l'émulation des ressources du matériel.

1.3.3.1 La para-virtualisation PV

La para-virtualisation (PV) est une technique de virtualisation efficace et légère qui à l'origine a été introduite par Xen, et adoptée plus tard par d'autres plates-formes de virtualisation. La PV ne nécessite pas d'extensions de virtualisation depuis le processeur de l'hôte. Toutefois, les clients para-virtualisés requièrent un noyau apte à la para-virtualisation ainsi que des drivers, de sorte que les clients soient conscients de l'hyperviseur et peuvent fonctionner efficacement sans émulation du matériel. Les noyaux permettant la PV existent pour Linux, NetBSD, FreeBSD et OpenSolaris. Il est à noter que le noyau Linux est compatible avec la para-virtualisation depuis sa version 2.6.24. En pratique, cela signifie que la para-virtualisation fonctionnera avec la plupart des distributions Linux (à l'exception de très vieilles).

1.3.3.2 La Full virtualisation HVM

La virtualisation complète HVM utilise des extensions de virtualisation du processeur de l'hôte pour virtualiser les domaines invités. La HVM nécessite les extensions matérielles Intel VT ou AMD-V. Xen utilise Qemu pour émuler le matériel, y compris le BIOS, le contrôleur de disque IDE, l'adaptateur graphique VGA, le contrôleur USB, les cartes réseau, etc. La HVM utilise des extensions matérielles pour améliorer les performances de l'émulation. Les domaines entièrement virtualisés avec la HVM ne nécessitent pas de support du noyau. Cela signifie que les systèmes d'exploitations tel que Windows peuvent être utilisés comme DomU sur Xen en HVM. Il est important de noter que les domaines invités pleinement virtualisés sont généralement plus lents que les clients para-virtualisés, en raison de l'émulation nécessaire.

1.4 Les bienfaits de la virtualisation et ses limites

Le passage à la virtualisation est une décision lourde en conséquence sur l'infrastructure d'un réseau. Cela ne dissuade pas pour autant les industriels de l'IT de franchir le pas. La virtualisation répond à des problématiques réelles pour relever les nouveaux défis auxquels l'informatique fait face : besoin de plus de

souplesse, d'évolutivité et d'isolation. La virtualisation en somme présente une multitude d'avantages qui ont motivé notre travail, elle présente néanmoins quelques inconvénients qui peuvent être surmontés.

1.4.1 Les bienfaits

Souvent, quand il s'agit d'argumenter en faveur de la virtualisation, l'un des avantages principaux qui apparait le plus souvent est la réduction des coûts de l'entreprise. Nous allons voir dans cette section que les avantages sont biens plus nombreux et souvent bien plus intéressants que la simple réduction des machines physiques utilisées.

- **Coûts réduits** : l'un des principaux avantages de la virtualisation est qu'elle nécessite moins de matériel pour exécuter le même type et la même quantité de logiciels, ce qui fait baisser les coûts globaux. La virtualisation permet une réelle consolidation du réseau en utilisant au mieux les ressources disponibles.
- **Redéploiement rapide et continuité de service** : la récupération des données simplifiée est un autre grand avantage de cette technologie. Par exemple, si le serveur virtuel devient soudainement corrompu il suffit de le supprimer et de le restaurer à partir de sa sauvegarde virtuelle. Cela représente un gain de temps et d'efforts considérable en comparaison à la méthode traditionnelle qui consiste à rétablir l'ensemble du système 'from scratch', puis de restaurer à partir de la dernière sauvegarde. Ainsi, un système virtuel corrompu peut être récupéré en quelques minutes.
- **Le testing** : La virtualisation fournit une plate-forme sûre sur laquelle il est permis de tester différentes configurations logicielles et sur différentes plateformes avant le déploiement. Ceci permet aux chercheurs par exemple de bricoler avec le logiciel jusqu'à obtenir ce qu'ils désirent exactement sans endommager par inadvertance le réseau existant.
- **Ecologique** : La baisse de la consommation d'énergie puisqu'on fait usage de moins de matériel informatique pour accomplir le même type de travail. Ceci permet de réduire les émissions de carbone, et d'être plus respectueux de l'environnement. Il est à noter que cet argument est devenu synonyme

d'image de marque des grandes entreprises de l'IT qui publient chaque année leurs empreintes carbone.

- **Sécurité et fiabilité** : la démarcation de la couche physique et de la couche logicielle permet d'améliorer la sécurité du système et sa fiabilité. Les systèmes virtualisés ne sont pas sujets à la corruption des pilotes de périphériques ou à des problèmes de mémoire.
- **La migration** : du fait de son aspect logiciel, la machine virtualisée peut être déplacée d'un emplacement à un autre en un rien de temps. Il suffit de s'assurer que le nouveau substrat physique de migration possède toutes les ressources physiques dont a besoin la machine virtuelle.

1.4.2 Les limitations

La virtualisation est une technologie à double tranchant, elle connait aussi un bon nombre d'inconvénients.

- **Les machines virtuelles reposent sur un substrat physique** : même si rares, les défaillances physiques peuvent être dévastatrices. Par exemple, si le disque dur principal qui contient toutes les données virtuelles et physiques est soudainement volé, brûlé, brisé ou endommagé alors tous les serveurs à la fois virtuels et physiques devront être restaurés. Il est ainsi conseillé d'ajouter de la redondance dans les infrastructures virtuelles, surtout quand il s'agit de machines virtuelles contenant des informations critiques.
- **Nécessite du matériel puissant** : la virtualisation est essentiellement dépendante de la puissance de calcul et de la mémoire. Ainsi, il faut prendre en compte à la fois beaucoup plus de mémoire et de puissance de traitement.
- **Nécessite une expertise pour la maintenance** : lorsqu'un problème survient dans un système virtualisé, cela nécessite dans certains cas un dépannage complexe. Malgré les outils mis à disposition, la complexité ajoutée par la virtualisation nécessite l'intervention d'un expert avisé.
- **L'overhead** : Il est tout à fait instinctif de penser à l'impact sur les performances qu'aura une infrastructure virtualisée par rapport à une infrastructure physique. En effet le système virtualisé, indépendamment du

temps de calcul requis par les machines virtuelles, aura besoin également d'un temps de calcul pour s'autogérer. Ainsi, notre hyperviseur Xen introduit un overhead de 3% alors que des hyperviseurs tel que KVM ont des overhead aux alentours des 20% [8]. De façon générale, l'usage de machines physiques puissantes avec beaucoup de ressources permet d'amoindrir l'effet de l'overhead sur les machines virtuelles.

1.5 La virtualisation appliquée aux points d'accès Wi-Fi

Nous avons dans ce qui précède présenté la virtualisation de façon générale ainsi que le cas particulier de l'hyperviseur Xen que nous utiliserons dans le reste de notre travail. Dans cette section du chapitre nous allons exposer le concept de point d'accès Wi-Fi virtuel en expliquant le cheminement de nos idées jusqu'à la mise en place notre solution.

1.5.1 Motivations

L'accessibilité, la qualité de service, la souplesse et l'évolutivité de l'infrastructure sont des problématiques auxquelles sont confrontés les WISP (Wireless Internet Service Provider). Toutes ces problématiques dépendent directement des caractéristiques du déploiement du réseau Wi-Fi, il faut qu'il assure une couverture étendue et dense. Il ne faut pas oublier aussi que les liaisons entre clients et borne Wi-Fi se font en ondes hertziennes dans des bandes fréquentielles limitées. Un point d'accès Wi-Fi standard fonctionnant dans la bande des 2.4GHz ou des 5GHz ne peut en théorie accueillir que 128 clients simultanément et en pratique 30 au maximum à cause des interférences inter et intra canaux. La solution pour assurer une couverture Wi-Fi maximale est d'installer autant de points d'accès que possible. Un autre problème qui se pose est alors l'interférence entre les canaux, si on prend l'exemple de la bande des 2,4 GHz nous avons à notre disposition trois canaux parfaitement orthogonaux (1, 6, 11). L'utilisation du même canal dans une même zone de couverture Wi-Fi causera des interférences et réduira les performances globales du réseau. Comment dans ce cas élargir un parc de bornes Wi-Fi sans rajouter des interférences supplémentaires si tous les canaux sont déjà occupés ? Il faudrait

dans un sens utiliser les ressources physiques déjà présentes, donc utiliser une borne d'accès Wi-Fi déjà utilisée.

D'une manière plus générale, la cohabitation de façon dynamique des réseaux Wi-Fi grâce à la virtualisation apparaît comme étant une solution prometteuse pour découpler l'infrastructure des services d'accès -permettant ainsi l'urbanisation des points d'accès.

C'est de là que la notion de point d'accès Wi-Fi virtuel prend tout son sens. Nous allons dans ce qui suit expliquer comment sommes-nous parvenu à virtualiser les points d'accès Wi-Fi.

1.5.2 Discussion sur les points d'accès Wi-Fi virtuels

Le concept même de point d'accès Wi-Fi virtuel laisse perplexe aux premiers abords. Les bornes Wi-Fi physiques sont à tort considérées comme étant des équipements réseaux spécialisés où le software et le hardware sont indissociables. Nous avons pourtant vu dans ce qui précède que pour pouvoir créer une machine virtuelle, tout ce dont on a besoin est un substrat physique (machine hôte) qui possède les ressources nécessaires pour faire fonctionner cette machine virtuelle. Nous avons aussi vu que le concept de NFV stipule que les nœuds de réseau sont virtualisables. C'est avec cette optique que nous avons abordé le sujet dans notre thèse, et c'est ainsi que notre première réflexion a été de savoir : que doit-on fournir en termes d'I/O, de CPU, de RAM, de drivers, de cartes réseaux pour pouvoir créer une machine virtuelle remplissant en tous points les fonctionnalités d'un point d'accès Wi-Fi physique ? Nous savons d'ores et déjà qu'un point d'accès Wi-Fi virtuel sera une machine virtuelle lancée sur un point d'accès physique, et qui sera munis d'interfaces Wi-Fi virtuelles lui permettant de communiquer avec les interfaces physiques du substrat physique dans le but de communiquer sur le medium radio.

Pour pouvoir répondre à cette question nous nous pencherons, dans la sous-section qui suit, sur la compréhension du fonctionnement d'un point d'accès Wi-Fi physique pour mieux comprendre à quel niveau devons-nous intervenir.

1.5.3 Le fonctionnement d'un point d'accès Wi-Fi physique

Nous nous intéressons ici au fonctionnement d'un point d'accès Wi-Fi au niveau du hardware. Le principe général du fonctionnement de base d'un routeur Wi-Fi physique est le suivant : les paquets reçus sur les interfaces physiques sont mis en file d'attente au niveau de la carte avant d'être traités par le pilote qui réside dans le *user-space*. Ce pilote inspecte l'adresse de destination de chaque paquet, consulte la table de routage pour récupérer l'adresse du réseau de destination et achemine le paquet vers l'interface de sortie correspondante selon l'algorithme de routage choisi. La carte Wi-Fi physique installée au niveau du point d'accès doit être configurée en mode Access Point et doit être affectée d'un SSID (Service Set Identifier) qui l'identifie par les utilisateurs aux alentours.

D'un point de vu système nous pouvons décomposer le fonctionnement du Wi-Fi comme suit :

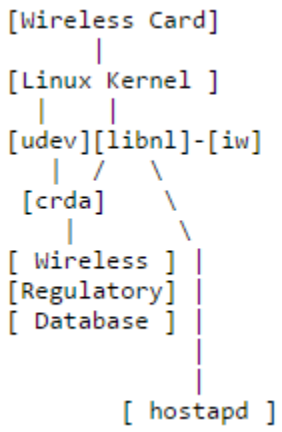

Figure 4: Diagramme d'interaction système du Wi-Fi

Les différentes composantes ont un rôle bien précis :

- Wireless Card : gère l'émission/réception des paquets dans le réseau sans fil.
- Linux Kernel : Le Linux Kernel contient le driver de la carte réseau sans fil, le sous-système mac80211 qui gère la génération de paquets et l'ordonnancement, et le sous-système nl80211, qui gère la configuration des interfaces sans fil pour l'user-space.

- libnl : libnl est la couche de transport utilisé pour communiquer avec le kernel via NetLink [27].
- udev : udev est un gestionnaire de périphérique utilisé par le kernel pour transmettre des événements / appels à crda.
- iw : iw est un utilitaire du user-space que nous pouvons utiliser pour vérifier que libnl fonctionne correctement, il permet aussi de créer des interfaces sans fil virtuels supplémentaires sur la carte sans fil.
- crda : crda est un programme du user-space que les requêtes du noyau utilisent (par udev) pour trouver quels canaux / fréquences sont utilisables, et à quelle puissance d'émission. Il déplace l'information sur des tables statiques maintenus dans le noyau vers le user-space, ce qui leurs permet d'être mis à jour sans recharger le driver ou le redémarrer.
- Wireless Regulatory Database : base de données des fréquences disponibles et des puissances d'émission autorisées utilisés par le crda.
- hostapd : il s'agit du daemon qui génère les beacons et les autres paquets transmis dans les réseaux sans fil, aussi bien que le chiffrement WPA-PSK, WPA2, etc. [45].

1.5.4 La solution proposée

1.5.4.1 Environnement matériel

Le substrat physique utilisé est un boitier commercialisé par l'entreprise VirtuOR : la MNetBox [40], il s'agit d'une machine industrielle conçu pour supporter les applications réseaux. Ce sont des boitiers comprenant une carte mère, une carte Ethernet, une carte Wi-Fi ainsi que les connectiques correspondantes.

Nous présentons ici les spécifications du boitier :

- Interface réseau: 4x Intel® 1 Gigabit LAN
- Processeur: INTEL NM10 + D2700
- Chipset Wi-Fi b/g/n MIMO 2x2
- Mémoire vive: 4 Go de RAM DDR3 SODIMM
- Dimensions: 255(L) x 156(l) x 36(H) mm

- Disque Dure : HDD 250GB 2"1/2
- Totalement silencieux (Fanless)

Figure 5: Borne Wi-Fi physique de développement

Ces boitiers comportent quatre interfaces Gigabit LAN et une interface console pour permettre d'y accéder avec un terminal de commande tel que Minicom sous linux.

1.5.4.2 Environnement logiciel

Nous avons dans les boitiers un environnement propriétaire VirtuOR qui repose sur une distribution légère adaptée à la virtualisation et basée sur le kernel linux 3.13 avec une surcouche Xen 4.3 qui reste l'une des seules versions supportées en 32 bits. Nous aurons aussi besoin pour la mise en place de notre solution de l'utilitaire *iw* pour la création des interfaces Wi-Fi virtuelles, du daemon *hostapd* pour la génération des trames 802.11, du package *nl80211* (ensemble d'API) pour établir les règles de communications entre le user-space et le kernel. Au niveau du kernel, nous aurons besoin de *cfg80211* qui est la nouvelle API de configuration des réseaux Wi-Fi linux qui remplace le 'Wireless Extensions' (wext) [46]. Le *cfg80211* a l'interface appropriée pour que la couche de *nl80211* puisse le configurer. Il est à noter que le *cfg80211* est déployé dans le kernel, alors que *nl80211* se trouve dans le user-space. Le framework *mac80211* est nécessaire dans le kernel pour s'interfacer entre le *cfg80211* et le driver de la carte Wi-Fi utilisé (ath9k dans notre cas). Et finalement, nous aurons besoin du package bridge-utils pour créer et gérer nos ponts réseau [47].

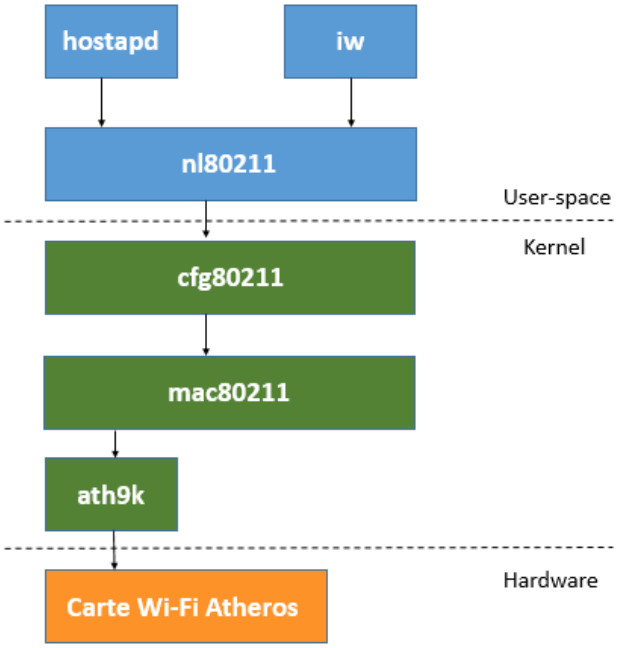

Figure 6: Interaction logicielle entre user-space et kernel

La Figure 6 résume l'environnement logiciel qu'on vient de décrire ci-dessus. Pour qu'un tel dispositif puisse communiquer de bout en bout, le user-space doit pouvoir interagir avec le kernel, d'où la nécessité d'installer *hostapd*, *iw* et *nl80211* dans le Dom0 (domaine privilégié) pour pouvoir traverser la couche d'hypervision. Le Dom0 se chargera par la suite de relayer le flux de données aux différentes machines virtuelles.

1.5.4.3 Le fonctionnement de la solution

Comme nous l'avons expliqué, le fonctionnement de la carte Wi-Fi reste inchangé. Notre apport réside en l'encapsulation dans une instance virtuelle (Dom0) des différents utilitaires communiquant avec le kernel, et à la création et l'utilisation d'interfaces Wi-Fi virtuelles pour l'émission et la réception des trames 802.11. Les interfaces virtuelles sont reliées aux interfaces physiques à l'aide d'une couche intermédiaire qui assure le dialogue entre l'interface virtuelle frontend [48] au sein de la machine virtuelle et le composant matériel. Au niveau de l'interface Wi-Fi physique, les paquets reçus sont mis dans une file d'attente pour être servis par la suite moyennant le pilote installé au niveau du domaine

privilégié Dom0. Chaque point d'accès virtuel dispose au moins d'une interface virtuelle liée par un pont réseau à une interface Wi-Fi physique. Il existe plusieurs façons de gérer les interfaces virtuelles, la plus simple étant le mode "bridge" [47]. Nous créons ainsi un bridge par interface Wi-Fi virtuelle créée grâce à la fonctionnalité VAP (Virtual Access Point). En effet, certains pilotes Wi-Fi offrent la possibilité de partager la carte réseau avec la fonction multi-SSID grâce à un multiplexage temporelle entre plusieurs interfaces d'accès. Chacune de ces interfaces VAP est paramétrée différemment selon le besoin. Toute interface VAP est reliée à une machine virtuelle qui joue le rôle d'un point d'accès virtuel à travers un bridge dédié. Le domaine privilégié Dom0 assure le dialogue avec l'hyperviseur qui se charge de l'interfaçage entre le matériel et les instances de points d'accès virtuels par les mécanismes de transfert de donnés inter-domaines.

Le domaine privilégié joue le rôle de gestionnaire d'interfaces Wi-Fi virtuelles à l'aide du pilote ath9k [49]. Ce dernier offre la possibilité de partager la carte Wi-Fi physique en plusieurs interfaces VAP. C'est un pilote Linux libre pour les cartes Wi-Fi 802.11b/g/n basé sur une carte Atheros. Ce type de pilote offre la capacité à utiliser la carte en tant qu'une simple interface réseau ou la possibilité de créer une multitude de cartes réseaux d'accès VAP, limité à quatre dans notre cas. Chacune des interfaces VAP partage la ressource physique Wi-Fi avec des SSID ainsi que des adresses MAC différentes. Un mécanisme de multiplexage/démultiplexage permet d'acheminer les paquets entre les interfaces virtuelles et les interfaces physiques. Le driver ath9k offre des outils puissants pour la configuration des cartes Wi-Fi permettant de configurer plusieurs modes de fonctionnement tels que le mode AP (Access Point), Ad-hoc, Mesh, etc. Les points d'accès virtuels sont des machines virtuelles. Ils contiennent un système d'exploitation Linux modifié pour supporter la para-virtualisation et des interfaces virtuelles dont une au moins est liée à l'accès Wi-Fi. Le noyau de son système d'exploitation est basé également sur un kernel Linux modifié afin d'être allégé au maximum. Toutefois, il fournit les modules nécessaires pour le bon fonctionnent de l'instance en tant que point d'accès. Le driver de Xen de l'instance frontend driver est chargé de l'émission/réception des flux réseaux sur ses différentes interfaces, principalement, l'interface qui la relie à la carte Wi-Fi physique.

Pour assurer le routage, un routeur logiciel *Quagga* [50] est installé sur ces points d'accès virtuels. La configuration du routeur est susceptible d'offrir par

exemple une sortie vers le réseau internet et réaliser les configurations de routage adaptées à chaque type de flux. L'affectation d'adresse IP (IPv4 ou IPV6) aux clients associés à un réseau sans fil virtuel est assurée par un serveur DHCP installé sur chaque point d'accès virtuel. La Figure 7 illustre l'architecture des points d'accès Wi-Fi virtuels présentés dans cette partie.

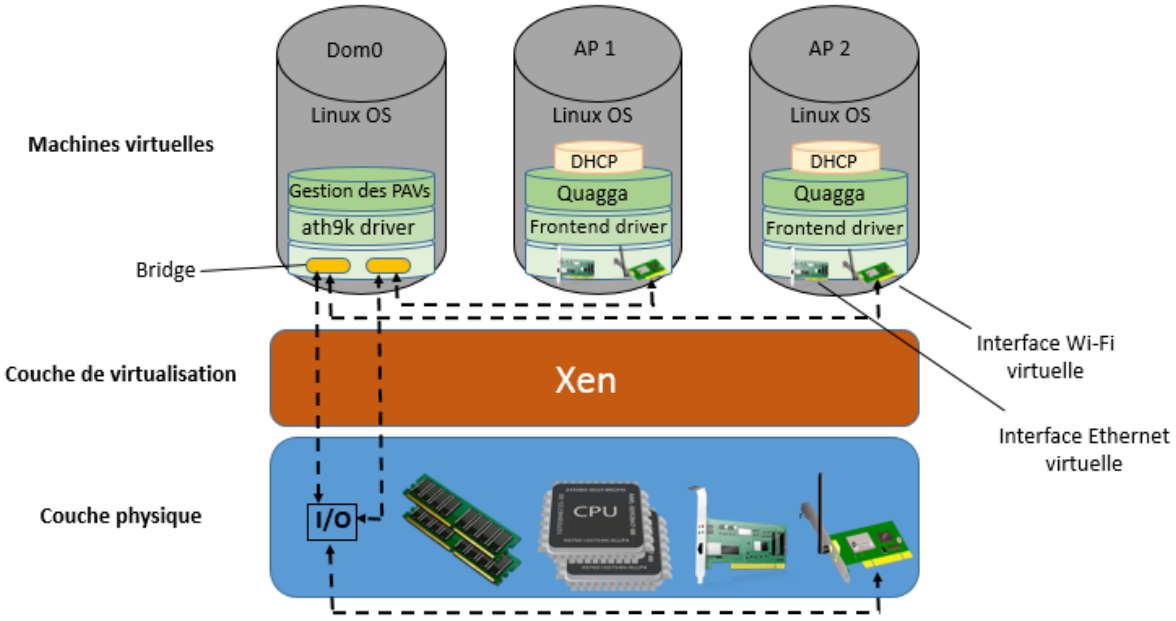

Figure 7: Architecture des points d'accès Wi-Fi virtuels

1.5.5 La mise en marche des points d'accès Wi-Fi virtuels

Une fois que l'environnement logiciel est prêt, le point d'accès n'est pas encore activé, il nous faut configurer puis 'allumer' le medium radio. Nous pouvons configurer les paramètres des points d'accès qu'on désire utiliser à l'aide du daemon *hostapd*. Le *hostapd* permet la création d'un point d'accès Wi-Fi à travers un fichier de configuration nativement présent sous */etc/hostapd/hostapd.conf*. Ce fichier contient tous les paramètres de configuration requis pour un point d'accès Wi-Fi (numéro de canal, SSID, type d'authentification, maximum de client, intervalle entre beacons, configuration du MIMO etc.). Nous pouvons lancer autant de daemon *hostapd* qu'il existe d'interfaces Wi-Fi virtuelles. Nous devrons dans ce cas-là dupliquer le fichier de

configuration *hostapd.conf*, et configurer chaque point d'accès Wi-Fi virtuel indépendamment l'un de l'autre. Il est important de mentionner dans chaque fichier de configuration quel bridge et quelle interface Wi-Fi virtuelle *hostapd* utilisera pour communiquer avec la carte Wi-Fi physique.

Une fois les fichiers *hostapd.conf* configurés, il suffit de lancer *hostapd* autant de fois que de nombre de points d'accès Wi-Fi nous voulons créer. Par exemple, si nous avons trois points d'accès Wi-Fi virtuels PAV1, PAV2 et PAV3, dont les fichiers de configurations sont respectivement etc/hostapdPAV1.conf, etc/hostapdPAV2.conf et etc/hostapdPAV3.conf, nous lancerons les commandes suivantes :

```
#> hostapd hostapdPAV1.conf
#> hostapd hostapdPAV2.conf
#> hostapd hostapdPAV3.conf
```

Les trois SSID apparaissent alors chez n'importe quel client Wi-Fi à proximité. Les trois points d'accès Wi-Fi virtuels sont intrinsèquement différents et indépendants les uns des autres. Chaque point d'accès ainsi crée peut supporter différentes couches protocolaires et ainsi créer un réseau Wi-Fi qui lui est propre et cohabitant avec les autres points d'accès.

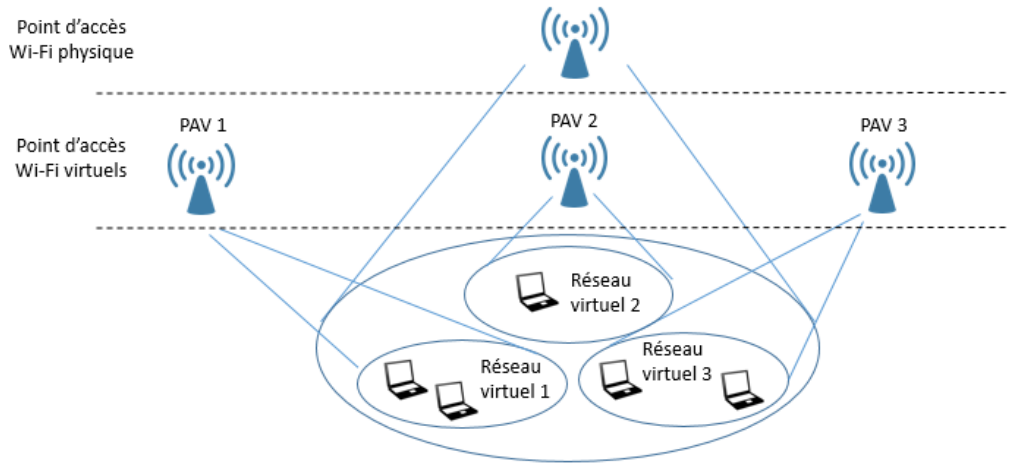

Figure 8: Réseaux Wi-Fi virtuels

1.5.6 Expériences

Notre banc d'essai expérimental est constitué de quatre machines physiques. Un point d'accès physique fournit les ressources virtuelles nécessaires pour instancier trois points d'accès virtuels. Trois ordinateurs portables considérés comme des clients (client 1, client 2 et client 3) sont reliés respectivement à leur point d'accès Wi-Fi virtuel. Chacun de ces ordinateurs possède un processeur C2D 1,6 GHz, 2 Go de mémoire RAM et une interface réseau Wi-Fi à 54 Mbit/s de débit. Les machines utilisées pour générer le trafic sont dotées de cartes variées de Wi-Fi. Le nœud physique qui sera l'hôte de tous les points d'accès virtuels est le boitier décrit dans l'environnement matériel en section 1.5.4. Toutes les instances de points d'accès virtuels ont cette configuration : un processeur virtuel VCPU x86, deux interfaces virtuelles dont l'une est reliée à l'accès sans fil, une image disque de 20 Mo, 80 Mo de mémoire RAM, un serveur DHCP installé et *Quagga* comme routeur pour le trafic réseau. La configuration logicielle utilisée lors des tests est un hyperviseur Xen avec un noyau Linux adapté à Xen en tant que domaine Dom0 et le noyau Linux supportant Xen en tant que domaine virtuel. À partir de ces machines, nous récupérons les statistiques de la bande passante, la variation du délai et le taux de perte.

1.5.6.1 Expérience 1 : Emission/Réception d'un flux TCP

But de l'expérience : l'objectif de cette expérience est d'évaluer la bande passante disponible pour chaque point d'accès virtuel en variant le nombre total de machines instanciées. Nous mettons en évidence la concurrence sur la bande passante afin de juger l'efficacité du modèle proposé, évaluer l'équité du partage des ressources et la prédictibilité en termes de bande passante offerte par chaque point d'accès virtuel. Un autre objectif est d'évaluer les coûts en termes du processeur et de mémoire vive pour estimer le nombre limite de points d'accès virtuels qui peuvent être créés en présence d'une machine physique.

Mise en œuvre : Pour cette expérience, nous disposons d'une machine avec une interface Wi-Fi physique qui héberge trois points d'accès virtuels. Chacun génère un flux TCP vers sa destination afin d'évaluer le partage de la bande passante. Sur chaque point d'accès virtuel, la configuration du SSID et de

la plage d'adresse IP du réseau virtuel sont différentes. La bande passante est mesurée avec un seul point d'accès virtuel durant 3 minutes, puis un autre point d'accès est ajouté tout en mesurant les nouvelles valeurs. Après 3 minutes, le troisième point d'accès virtuel est instancié pour mettre en évidence la concurrence sur la ressource physique. Les flux TCP sont émis/reçus par des machines physiques connectées chacune respectivement à un réseau virtuel. Nous envoyons simultanément les flux TCP afin d'évaluer la ressource bande passante disponible.

Résultats : L'évolution de la bande passante tout en variant le nombre des points d'accès virtuels qui consomment les ressources physiques est représentée dans la figure 9. Chaque 3 minutes, nous ajoutons un point d'accès virtuel qui reçoit un flux TCP depuis une machine physique jusqu'à faire fonctionner 3 points d'accès virtuels simultanément. Ces mesures sont prises avec un, deux et trois points d'accès virtuels nommés wifirouter01, wifirouter02 et wifirouter03.

La Figure 9 montre qu'il y a une forte concurrence sur la bande passante. En effet, la bande passante mesurée avec un seul point d'accès virtuel est en moyenne égale à 24 Mbit/s. Lors du lancement de la deuxième instance du point d'accès virtuel, le partage de cette ressource est quasi équitable, environ 12 Mbit/s. Avec le lancement de la troisième instance, il y a une baisse significative de la bande passante disponible pour chacune des machines virtuelles. La bande passante offerte par chacune des trois machines virtuelles est réduite à 7 Mbit/s en moyenne.

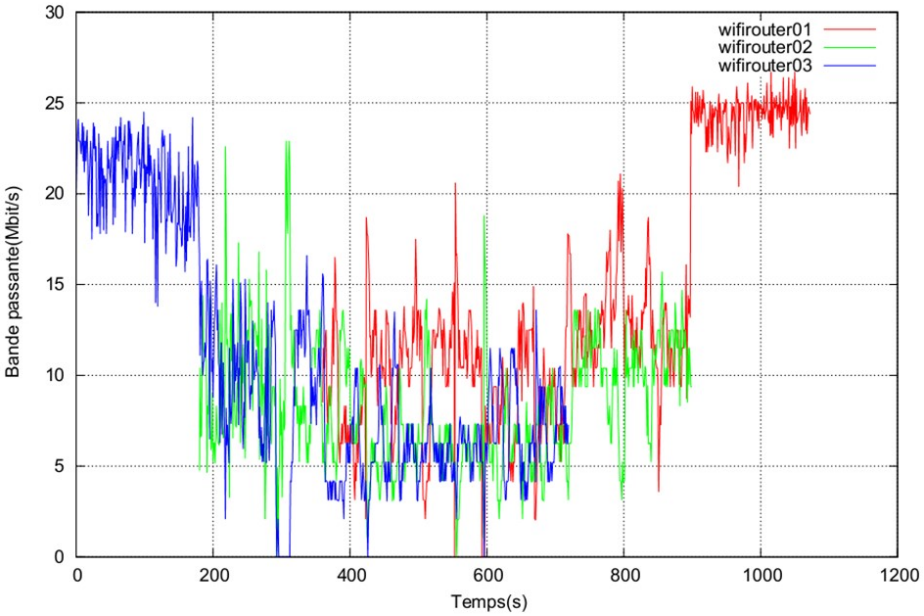

Figure 9: La répartition de la bande passante entre trois points d'accès Wi-Fi virtuels

D'après la figure 9, une fois qu'une instance libère la partie de la bande passante qu'elle utilise, cette portion de ressource est récupérée par les autres instances de point d'accès comme la montre la figure à l'instant 720s et 900s. En moyenne, la bande passante totale passe de 24 Mbit/s avec un seul point d'accès Wi-Fi virtuel, à 12 Mbit/s avec deux instances et enfin à 7 Mbit/s pour les trois simultanément. Il faut noter que les valeurs mesurées pour chaque point d'accès sont variables. Pour le coût en termes de processeur, la part de trois points d'accès est négligeable par rapport au domaine Dom0. Tandis que les machines virtuelles ne font que le routage des paquets dans leur propre réseau virtuel, le calcul du processeur est pratiquement réalisé au niveau du Dom0 qui est chargé de la gestion des interfaces d'accès VAP. De même pour la mémoire, nous remarquons une faible consommation de la mémoire au profit du domaine Dom0. En effet, le domaine privilégié effectue le transfert des paquets pour les différentes machines à sa charge.

Efficacité : La somme de la bande passante offerte par les machines virtuelles est comprise entre 22 et 24 Mbit/s, des valeurs très proches de celle du

système Linux sans virtualisation. Avec deux flux, le taux moyen total diminue peu pour un total de 21 Mbit/s. Enfin, avec les trois flux dans le même temps, le débit total est d'environ 20 Mbit/s. Nous pouvons donc affirmer l'efficacité du système mais avec un coût de calcul CPU plus important.

Equité : Le partage de la bande passante est quasi équitable entre les différents points d'accès virtuels. Par contre, le coût du calcul du processeur est très important pour le domaine privilégié Dom0. Toutefois, il est négligeable pour les autres machines virtuelles. On conclut que le partage de la bande passante est équitable, mais cette équité présente un certain délai de convergence. D'après la Figure 9, nous pouvons remarquer une phase transitoire durant laquelle une instance prend la moitié de la bande passante et les deux autres prennent le reste. Cette phase a duré presque 3 minutes, ce qui n'est pas négligeable. À la fin de cette phase, le partage de la bande passante devient très équitable.

Prédictibilité : La bande passante totale est toujours divisée en fonction du nombre des points d'accès Wi-Fi simultanés. En effet, la bande passante que peut offrir un point d'accès virtuel peut être estimée en fonction du nombre des instances qui l'ont précédée.

1.5.6.2 Expérience 2 : La variation de délai (la gigue)

Objectif de l'expérience : La deuxième expérience prend en considération la variation du délai pour les différents flux. Les réseaux virtuels sont adaptés aux caractéristiques des flux. Un film MPEG-2 pourrait être reproduit au niveau du récepteur avec un certain retard. Dans cette expérience, l'intérêt porte sur la gigue associée aux différents débits. La difficulté de la solution proposée provient de la difficulté à préciser la priorité entre les points d'accès Wi-Fi. Cela est dû à la stratégie d'isolement entre les instances des points d'accès virtuel. Les résultats sont décrits dans la figure 10.

Résultats : D'après la figure 10, nous pouvons constater que la gigue mesurée au niveau de trois points d'accès virtuels est en moyenne 1 ms. Les trois courbes ne dépassent pas les 5 ms pour chaque instance. Par contre, nous pouvons remarquer que les valeurs les plus hautes sont enregistrées au moment du lancement des trois différents flux MPEG-2 (flux UDP) à travers chaque point

d'accès. Après cette phase, les valeurs de gigue se stabilisent à une faible valeur d'environ 1 ms. Quelques pics de gigue pour deux machines virtuelles alors que la troisième machine garde la valeur ordinaire d'une 1 ms. Avec deux machines virtuelles et un flux UDP à 12 Mbit/s, nous remarquons que les valeurs de gigue baissent et se stabilisent à 0,5 ms en moyenne. Les valeurs les plus hautes ne dépassent pas 3 ms. De même, nous pouvons remarquer qu'une seule machine virtuelle enregistre des valeurs minimes de gigue. Concernant les trois flux de l'application MPEG-2, nous pouvons observer que la gigue des trois flux est comparable, car les instances sont configurées avec les mêmes privilèges. Si nous voulons donner quelques priorités à certains flux, nous devons le spécifier dans le nœud privilégié (Dom0) moyennant un algorithme permettant de prioriser les demandes d'accès d'une machine comme proposé en [51]. L'inconvénient de cette solution est la nécessité de connaître la nature du flux à traiter dans chaque point d'accès.

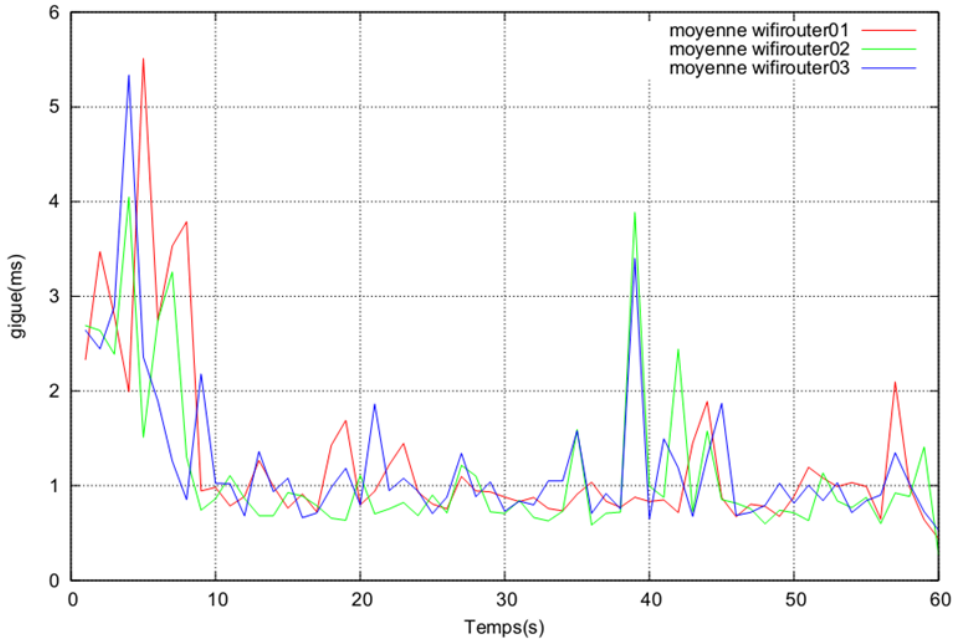

Figure 10: Répartition de la gigue entre trois points d'accès Wi-Fi virtuels

Efficacité : D'après cette expérience, nous constatons que la gigue augmente avec le nombre des machines virtuelles. L'efficacité du système diminue avec le nombre des instances de points d'accès virtuels en termes de variation du délai. À chaque fois que le nombre de point d'accès est abaissé par un, la moyenne de la variation du délai est divisée par deux.

Equité : La variation du délai est presque équitable avec 2 et 3 points d'accès virtuels. Mais parfois quelques machines virtuelles enregistrent un pic alors que les autres machines enregistrent des valeurs presque constantes. Nous concluons que la variation du délai est équitable.

Prédictibilité : En ce qui concerne la prédictibilité de variation du délai, nous pouvons affirmer une baisse de cet indicateur avec la diminution du nombre des points d'accès mais cette variation reste imprévisible.

1.5.6.3 Expérience 3 : Le taux de perte

Cette expérience reprend le même scénario précédent en utilisant trois points d'accès virtuels avec un flux UDP. Chacun émet à 8 Mbit/s. D'après la Figure 11, nous pouvons constater que le taux de perte mesuré au niveau des trois accès est en moyenne de 5%. Les trois courbes ne dépassent pas les 10% pour tous les points d'accès virtuels. Toutefois, le point d'accès virtuel, wifirouter02, enregistre des valeurs importantes au début de la phase de lancement des flux UDP. Avec des débits plus faibles, nous remarquons un taux de perte plus faible. Dans une autre expérience avec deux machines virtuelles et un flux UDP à 12 Mbit/s, nous remarquons que les valeurs de taux de perte baissent vers 4% en moyenne. De même, les valeurs les plus hautes ne dépassent pas 10% de perte.

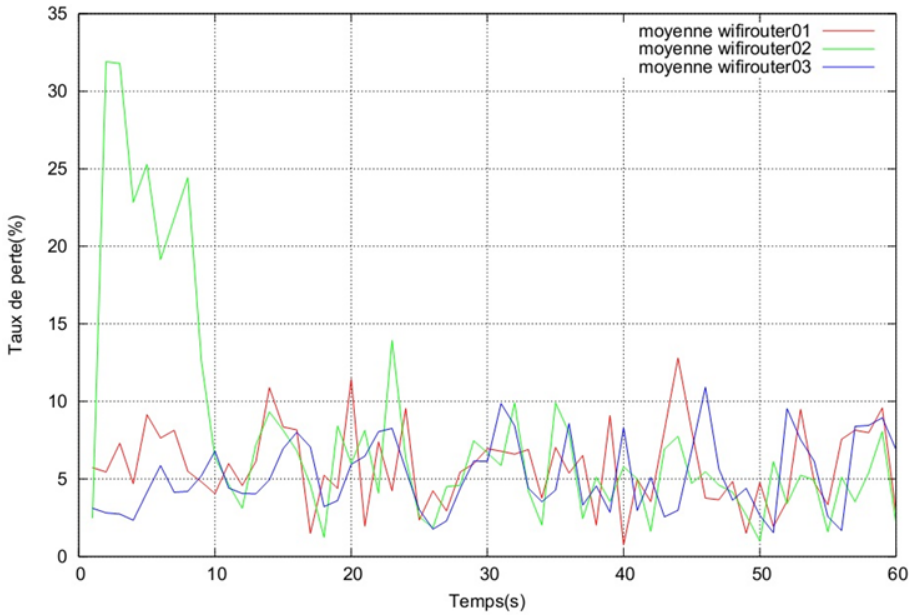

Figure 11: Variation du taux de perte avec trois points d'accès Wi-Fi virtuels avec un flux UDP

Le taux de perte mesuré avec un seul point d'accès virtuel est égal à la valeur de 2,5 % en moyenne. Les maximales ne dépassent pas le 5%. Les performances en termes de taux de perte deviennent médiocres avec l'augmentation de nombre des machines virtuelles et des débits des flux UDP utilisés.

Efficacité : D'après la figure 11, nous pouvons constater que le taux de perte dépend essentiellement du nombre des machines virtuelles et du débit des flux servis. En utilisant des flux UDP à des débits similaires, la bande passante totale est divisée par le nombre d'instances de point d'accès. Nous pouvons affirmer selon la courbe l'efficacité de la solution en termes de taux de perte.

Equité : Nous pouvons conclure que le partage de taux de perte est quasiment équitable avec 2 et 3 point d'accès virtuels.

Prédictibilité : En ce qui concerne la prédictibilité de taux de perte, nous pouvons estimer une baisse de ce taux avec la diminution du nombre des points d'accès.

1.5.7 Les cas d'utilisation

Un utilisateur Y d'un opérateur Oy se présente aux alentours de la MnetBox. L'opérateur Oy instancie son point d'accès APy au sein de la MNetBox et lui alloue un accès Wi-Fi VAP. L'utilisateur Y sélectionne depuis la liste des points d'accès Wi-Fi visible celui de son opérateur Oy. L'utilisateur Y s'associe au point d'accès virtuel de son opérateur. Une fois associé, le trafic de la communication entre le terminal de l'utilisateur Y et le point d'accès virtuel de son opérateur Oy est acheminé comme décrit dans la description technique de cette partie.

La gestion de la mobilité de l'utilisateur Y est prise en charge indépendamment de l'environnement radio dans lequel il est présent. S'il n'y a la présence que d'une seule MNetBox à laquelle l'utilisateur est connecté, en s'éloignant de la couverture radio qu'offre la box, Y passe de nouveau à son réseau opérateur cellulaire (dans le cas où le client est équipé d'un smartphone muni d'une carte SIM). Si nous sommes en présence de nombreuses MNetBox pour une couverture étendue, Y peut se déplacer d'une box à une autre de façon

transparente. L'avantage d'un point d'accès virtuel est sa mobilité. En effet, le point d'accès peut se déplacer avec l'utilisateur d'une box à une autre, à travers des mécanismes de migration de machines virtuelles. Nous partons de l'hypothèse que lorsque le signal radio d'une box franchit un seuil minimal, l'interface radio de Y scan les ressources disponibles, et envoie une demande d'association à la nouvelle box de manière transparente (nous verrons dans la section 3.3 du Chapitre 3 les problèmes liés à cette assertion). A ce moment-là, l'APy instancié par l'opérateur Oy migre d'une box à une autre à travers le DS (Distributed System) du réseau, pour garantir l'accès aux services de Y en tout lieu de couverture des MNetbox, et tout ceci sans rupture de service. La migration des machines virtuelles à chaud n'a pas été abordée lors de cette thèse, mais elle offre un potentiel intéressant qui mériterait d'être traitée dans des travaux avenirs.

1.6 Conclusion

Nous avons vu à travers le NFV, que l'utilisation d'un bridge par point d'accès virtuel nous permettait de créer de façon efficace des réseaux Wi-Fi virtuels. De par leurs indépendances, et leurs isolations, chaque point d'accès virtuel génère un réseau qui lui est propre et avec des spécificités intrinsèques indépendantes des autres VAPs : nous pouvons ainsi avoir sur une MNetBox utilisant une seule carte Wi-Fi, avec par exemple, un réseau ouvert, un autre protégé par clé WEP, un autre par une clé WPA-PSK, et un dernier avec du WPA2-Enterprise.

Cette modularité des points d'accès est tout à fait en cohésion avec ce que les opérateurs mobiles téléphoniques cherchent à établir : instancier à la demande un accès aux services requis par un client, puis libérer les ressources lorsque ce même client a fini d'accéder à son service. L'élasticité de notre solution technique laisse le champ libre à des scenarios riches et variés. A travers cet accomplissement technique nous avons présenté l'un des mécanismes de l'urbanisation des points d'accès Wi-Fi. De plus le NFV apporte une certaine souplesse à l'infrastructure du réseau Wi-Fi, permettant d'instancier à la demande des points d'accès. Nous avons évoqué dans la section 1.5.7 la possibilité pour les

opérateurs de créer un point d'accès Wi-Fi virtuel pour un client. Nous verrons en sous-section 2.3.4.2 dans le Chapitre 2 comment un client peut de lui-même initier ce processus d'instanciation à la demande.

Chapitre 2 : Hotspot2.0 virtuel et approvisionnement des droits d'accès

2.1 Introduction

Au fil des années, la couverture Wi-Fi s'est élargie pour offrir aux utilisateurs une connectivité dans de nouveaux environnements et sur une large gamme de périphériques clients. L'accès public aux hotspots, où l'infrastructure Wi-Fi est partagée entre tous les utilisateurs souhaitant se connecter au réseau, gratuitement ou non, est à présent répandu dans le monde entier, à partir d'un nombre croissant d'appareils mobiles équipés de cartes Wi-Fi. Le nombre de bornes Wi-Fi a rapidement augmenté ces dix dernières années, passant initialement des petits réseaux indépendants, qui ne couvraient souvent qu'un seul café ou hôtel, aux réseaux importants des fournisseurs, conçus pour fournir un accès sans fil dans des lieux publics très fréquentés. La connectivité Wi-Fi dans les lieux publics gagnant en disponibilité et en popularité, doit fournir un accès transparent aux réseaux, avec des connexions sécurisées et un service ininterrompu. Le Wi-Fi est devenu essentiel pour les fournisseurs, afin d'ajouter de la valeur à leurs offres de services et d'améliorer l'expérience de l'utilisateur.

Il y a moins d'une décennie, les réseaux mobiles ont été fondés uniquement sur le spectre sous licence pour permettre l'accès aux données à l'utilisateur. La tendance a changé depuis, et les opérateurs de téléphonie mobile ont élaboré une feuille de route intégrant l'accès au réseau sur le spectre non licencié en utilisant le Wi-Fi. Le principal facteur de ce changement a été la forte demande pour la vidéo et d'autres services de données à haut débit qui ont submergé les réseaux 3G/4G. La plupart des prévisions des opérateurs basées sur la consommation des services de données au cours des prochaines années, aboutissent à la conclusion que les besoins futurs de données ne pourront pas être satisfaits par des améliorations au niveau du réseau cellulaire traditionnel. Il en résulte qu'un intérêt croissant a été porté pour la technologie Wi-Fi, maintenant disponible sur tous les nouveaux smartphones et autres équipements mobiles. Le

Wi-Fi offre une connexion à haut débit, et est généralement accessible à la maison ou au travail, qui sont les deux lieux où la plupart de la consommation de données mobiles a lieu. A ces lieux privés, nous pouvons ajouter les lieux publics et semi-publics tels que les hôtels et les centres de congrès, les stades et les aéroports ainsi que les cafés. Les opérateurs mobiles connaissent un besoin croissant d'utiliser ces infrastructures Wi-Fi assurant l'acheminement du trafic client. En outre, le point d'accès Wi-Fi permet de délester les données du réseau 3G/4G de l'opérateur et permet aussi d'assurer l'itinérance pour les abonnés des autres opérateurs. Pour un usage public sans abonnement existant, un portail captif sur le point d'accès Wi-Fi permet à un utilisateur de saisir ses coordonnées bancaires pour une utilisation à court terme. L'opérateur hotspot peut vendre l'infrastructure Wi-Fi aux entreprises proposant des services publics, tout en maintenant son propre réseau d'accès. Bien qu'il soit possible aujourd'hui d'offrir un service de hotspot Wi-Fi complet à partir d'un réseau public ou public-privé (à double usage, tel que la plupart des boxs internet en France), il existe encore des obstacles à l'adoption généralisée. Les normes Wi-Fi existantes ainsi que les logiciels de gestion de connexion d'un appareil mobile n'ont pas été développés dans l'optique du hotspot. Il n'est donc pas surprenant que les services actuels diffèrent d'un opérateur à un autre et exigent l'intervention de l'utilisateur.

Par analogie, les téléphones cellulaires, quand ils ne peuvent pas trouver leur réseau domestique, s'identifient et s'enregistrent automatiquement chez l'un des partenaires nationaux ou internationaux sans la nécessité d'une intervention de l'utilisateur. À ce jour, le Wi-Fi a manqué d'un protocole pour rationaliser cette fonction. Il est bien sûr possible de configurer un appareil mobile Wi-Fi pour permettre un roaming automatique sur le hotspot visité, mais cette manipulation reste assez complexe et loin d'être universelle. Les points d'accès Wi-Fi d'aujourd'hui n'ont qu'un seul identifiant accessible au public, le SSID. Ainsi, ce SSID doit être utilisé pour indiquer différents types de réseaux. La plupart des SSID reflètent l'organisation d'exploitation du point d'accès local, comme "Starbucks" ou "LIP6-guest", tandis que d'autres indiquent l'accès à un fournisseur de services, "SFR WiFi Mobile". Quand un appareil mobile cherche un point d'accès Wi-Fi, il a deux options, soit il prend une liste de SSID configurés en amont comme «LIP6-guest» et cherche une correspondance, soit il tente de s'associer à chaque SSID ouvert et effectue des tests pour voir s'il peut accéder à

Internet. Dans le premier cas, il peut rater des réseaux auxquels il est légitimement autorisé mais qu'il ne connait pas (non configurés), tandis que le second cas prend beaucoup plus de temps, consomme de l'énergie et soulève des questions de confidentialité et de légalité.

Avec la nouvelle certification Passpoint de la Wi-Fi alliance, l'information sur les services et les fournisseurs de services d'un hotspot deviennent accessibles indépendamment du SSID. Un nouveau protocole permet à l'appareil mobile de découvrir un profil complet du hotspot avant qu'il ne s'y associe, de sorte qu'il puisse rapidement identifier et prioriser les hotspots adaptés à ses besoins. L'utilisation des noms des fournisseurs de services sans ambiguïtés simplifie la tâche à l'appareil mobile pour reconnaitre les points d'accès. Avec Passpoint, le mobile peut identifier les points d'accès appropriés et sélectionner le meilleur tout en restant dans la poche de l'utilisateur. Il peut alors s'authentifier automatiquement et commencer à utiliser le service tout en étant protégé par la sécurité sous-jacente.

Nous allons ainsi dans ce chapitre nous pencher de plus près sur ces nouveaux mécanismes permis par Passpoint ainsi que les enjeux qu'ils représentent. Nous verrons ensuite comment nous avons procédé pour virtualiser une telle technologie. Nous aborderons en deuxième partie de ce chapitre l'architecture que nous avons proposée pour pouvoir assurer une sécurité accrue de bout en bout dans ces points d'accès Wi-Fi de nouvelle génération.

2.2 La nouvelle génération de réseaux Wi-Fi: Hotspot 2.0

2.2.1 Historique

Hotspot 2.0, également connu sous le nom de « HS2 », « Wi-Fi CERTIFIED Passpoint » ou « Passpoint », est une nouvelle approche de l'accès Wi-Fi public élaborée par la Wi-Fi Alliance. Hotspot2.0 automatise le processus de sélection, d'association et d'authentification au réseau, ce qui permet une connexion transparente entre les hotspots et les appareils mobiles, tout en offrant un très haut niveau de sécurité WPA2. Hotspot2.0 est souvent considéré comme la

technologie qui permettra une expérience de type cellulaire dans les réseaux Wi-Fi.

HS2 a été élaborée à travers des releases successives. La toute première en juin 2012 spécifiait les différentes normes et protocoles de base (notamment l'amendement IEEE 802.11u) pour découvrir et sélectionner les réseaux Wi-Fi au travers de nouvelles trames 802.11 GAS/ANQP. La release 2 a été publié en octobre 2014 et fournit de nouveaux éléments de réponse quant au « online sign-up » et la priorisation des points d'accès Wi-Fi HS2. Passpoint a été développé par la Wi-Fi Alliance à travers des partenariats internationaux avec les fabricants d'équipements mobiles, les fournisseurs d'équipement de réseau et les opérateurs. Cette collaboration tend à standardiser l'accès aux points d'accès Wi-Fi pour permettre la meilleure expérience possible aux utilisateurs finaux.

2.2.2 Les normes et protocoles du Hotspot2.0

Le Hotspot2.0 repose sur plusieurs protocoles qui assurent une nouvelle approche de sélection et découverte de réseau, et une sécurité accrue. Nous décrivons ces protocoles ci-dessous.

2.2.2.1 IEEE 802.11u

802.11u (publié le 25 Février 2011) est un amendement à la norme IEEE 802.11-2007 pour ajouter des fonctionnalités qui améliorent l'interfonctionnement avec les réseaux externes [52]. L'une des fonctionnalités primordiale du 802.11u est appelée «Network discovery and selection » et apporte les innovations suivantes :

- L'AP fournit une liste d'informations aux clients mobiles tels que, les partenaires de roaming, les informations concernant la zone visitée, le nom de domaine, le type de réseaux (public, privé, payant, gratuit), etc. Ces informations sont primordiales à l'équipement mobile désirant se connecter à l'AP et sont échangées sous un nouveau format de messages qu'on appelle GAS/ ANQP.
- Le Generic Advertisement Service (GAS), qui fournit pour la couche 2 un protocole de transport des trames d'informations entre le client et l'AP avant l'authentification.

- L'Access Network Query Protocol (ANQP), qui est un protocole de requête et de réponse utilisé entre un appareil mobile et un AP pour découvrir un ensemble d'informations, y compris le nom de domaine de l'opérateur du hotspot (un élément unique au monde), les partenaires d'itinérance accessible via le hotspot, la méthode EAP pris en charge pour l'authentification, le type d'adresse IP disponible (IPv4, IPv6), et d'autres métadonnées utiles dans le processus de sélection de réseau d'un appareil mobile.

Le 802.11u introduit la notion de fournisseur de services d'abonnement (Subscription Service Provider : SSP), qui est l'entité responsable de la gestion de l'abonnement de l'utilisateur et les informations d'identification associées. Plusieurs SSPs sont représentés sur un seul et même AP du fait des accords d'itinérances signés entre eux, ainsi un opérateur partenaire de l'opérateur du client mobile pour accueillir ce dernier. Le concept de SSP est important car il rompt la relation jusqu'alors faite entre SSID d'un AP et le droit d'accès. L'itinérance devient plus facile car le client maintenant demande quel SSP est accessible sur l'AP, ce qui ne donne plus de significativité au SSID.

Dans la figure 12, nous représentons schématiquement l'échange qui se fait entre un client et un AP. Cette phase de pré-sélection représentative du standard 802.11u s'effectue pendant que le client se trouve encore connecté à son réseau cellulaire de données. Une fois que l'échange de messages ANQP (encapsulés dans des trames GAS) est concluant et que l'équipement mobile décide de se connecter à l'AP, la phase d'authentification aura lieu.

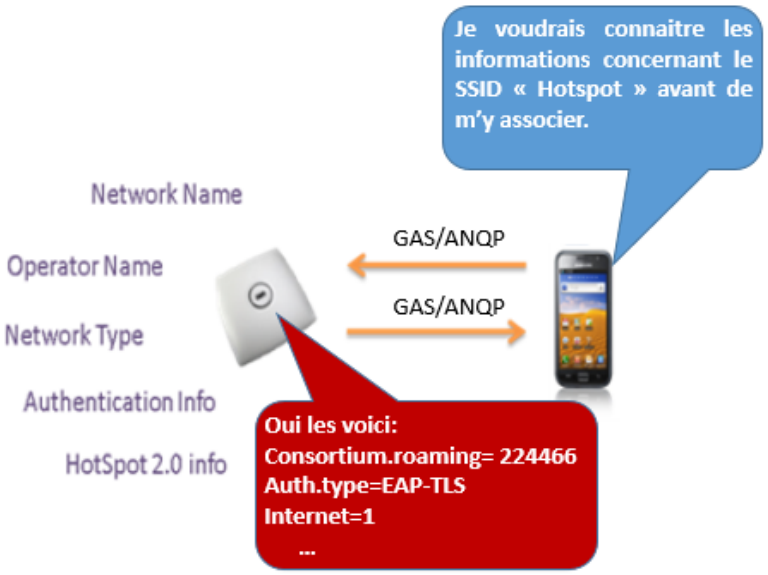

Figure 12 : Echange GAS/ANQP

2.2.2.2 IEEE 802.11i

IEEE 802.11i définit un réseau de sécurité robuste (Robust Security Network ou RSN) comportant des améliorations par rapport au mode de sécurisation WEP (Wired Equivalent Privacy) préconisé par le standard IEEE 802.11 [53]. Cet amendement a augmenté les méthodes d'authentification et de chiffrement. La norme utilise les moyens d'authentification et de chiffrement suivants :

- Le standard IEEE 802.1x, qui permet de réaliser une authentification par contrôle de l'accès réseau au port (Port Based Network Access Control) et autoriser ou non une liaison physique entre un client et un point d'accès.
- Le processus de chiffrement basé sur l'algorithme AES (Advanced Encryption Standard) permettant de réaliser le mode de sécurisation WPA2. La confidentialité des données (cryptage) est fournie en utilisant le CCMP (Counter-Mode/CBC-Mac protocol) qui utilise le cryptage AES sur 128 bits et offre une résistance cryptographique similaire à des réseaux cellulaires 3G.

- Le processus de chiffrement TKIP (Temporal Key Integrity Protocol) permettant d'obtenir le mode de sécurisation WPA (moins robuste que WPA2) à l'aide de clés de 128 bits dynamiques modifiées de manière aléatoire.

Hotspot2.0 repose sur ce qu'on appelle 802.11i WPA2-Enterprise qui exclut l'utilisation du chiffrement TKIP. IEEE 802.11i WPA2-Enterprise définit une norme pour établir un canal sécurisé, avec une authentification mutuelle entre le client Wi-Fi et le point d'accès avant la mise en place de toute communication IP. L'authentification mutuelle est effectuée en utilisant EAP (Extensible Authentication Protocol) sur IEEE 802.1X en utilisant un serveur d'authentification tel que RADIUS. Dans ce qui suit, nous présentons brièvement les protocoles 802.1X et RADIUS qui sont deux éléments importants dans la compréhension du processus d'authentification du Hotspot2.0.

2.2.2.3 802.1X

Dans la norme 802.1X, les équipements aux frontières (AP) sont en charge d'initier les sessions d'EAP avec les équipements tentant de s'associer avec eux. Il est important de mentionner que les protocoles IEEE 802 sont des protocoles de bas niveau. Par conséquent, les messages EAP sont directement envoyés sur la liaison physique encapsulés à l'intérieur du protocole correspondant. Dans notre cas, 802.1X va être mis en œuvre sur le Wi-Fi pour que l'équipement mobile qui désire s'associer au point d'accès puisse s'authentifier. Le client mobile et le point d'accès transmettent les messages EAP dans les trames Wi-Fi directement jusqu'à la fin de la session EAP. Au final, une décision d'acceptation ou de rejet est prise par l'AP pour donner l'accès au client dépendamment de sa légitimité sur ce réseau. Comme les messages EAP transportent des requêtes de droits d'accès, ils doivent être transmis à un serveur d'authentification qui est capable d'effectuer le processus d'authentification correctement. La solution la plus courante consiste à encapsuler les paquets EAP à l'intérieur du protocole RADIUS pour acheminer les paquets à travers un réseau de couche 3. RADIUS est détaillé dans la section suivante.

802.1X est mis en œuvre directement sur la couche de liaison de données, il en résulte que des périphériques non authentifiés n'ont pas accès au cœur du réseau. Par exemple, ils ne peuvent pas récupérer une adresse IP à partir du serveur DHCP ou interroger le serveur de nom de domaine. Ainsi, tout le trafic passant par un point d'accès Wi-Fi implémentant 802.1X sera surement du trafic provenant de clients autorisés et authentifiés. Les échanges de messages EAP se font à travers une encapsulation 802.11 appelée EAPOW (EAP Over WLAN). A son tour, la borne Wi-Fi doit être en mesure d'encapsuler les messages à l'intérieur des paquets RADIUS et de les transmettre à son serveur d'authentification (cette fois-ci, il s'agit d'une communication à travers un câble). Jusqu'à ce que l'authentification réussisse, l'appareil est incapable d'accéder à d'autres ressources sur le réseau.

Avant l'authentification, seule l'adresse MAC du client est disponible à la borne Wi-Fi. Cette information suffit pour construire les règles de firewalling adéquates pour permettre l'accès aux clients. Lorsque le point d'accès Wi-Fi encapsule les messages EAP dans des paquets RADIUS, il ajoute un ensemble d'informations sur lui-même et sur le client en cours d'authentification. Bien qu'aucune norme n'existe sur les informations obligatoires, il est de rigueur d'ajouter l'adresse MAC des deux équipements, l'adresse IP de l'AP et le SSID.

Les méthodes d'authentifications EAP préconisées par Hotspot2.0 sont résumées dans le tableau 1:

Table 1 : Les méthodes EAP recommandées par Hotspot2.0

Type d'identifiant	Méthode EAP
Certificat X.509	EAP-TLS
SIM/USIM	EAP-SIM, EAP-AKA
Username/password (avec certificat du côté serveur)	EAP-TTLS with MSCHAPv2

Ces différentes méthodes EAP préconisées par HotSpot2.0 ont été choisies pour recouvrir tous les clients mobiles possibles :

- Tous les clients mobiles sont en mesure de supporter l'EAP-Transport Layer Security (TLS) comme définit dans la RFC 5216 en utilisant des certificats digitaux X.509 [54].

- Si un client mobile a une carte SIM (Subscriber Identity Module), alors EAP-SIM peut être utilisé comme définit dans la RFC 4186 [55].
- Si le client mobile a une carte SIM UMTS (USIM), alors EAP-AKA (Authentication and Key Agreement) peut être utilisé comme définit dans la RFC 4187 [56].
- Tous les clients mobiles sont en mesure de supporter l'EAP-TTLS (Tunneled Transport Layer Security) comme défini dans la RFC 5281 [57] avec MS-CHAPv2 défini dans la RFC 2759 [58] qui utilise un jeu de username/password niveau client et un certificat côté serveur.

La Figure 13 présente une authentification 802.1X-EAP générale (toutes les méthodes EAP se basent sur ce schéma) entre un équipement mobile et une borne Wi-Fi.

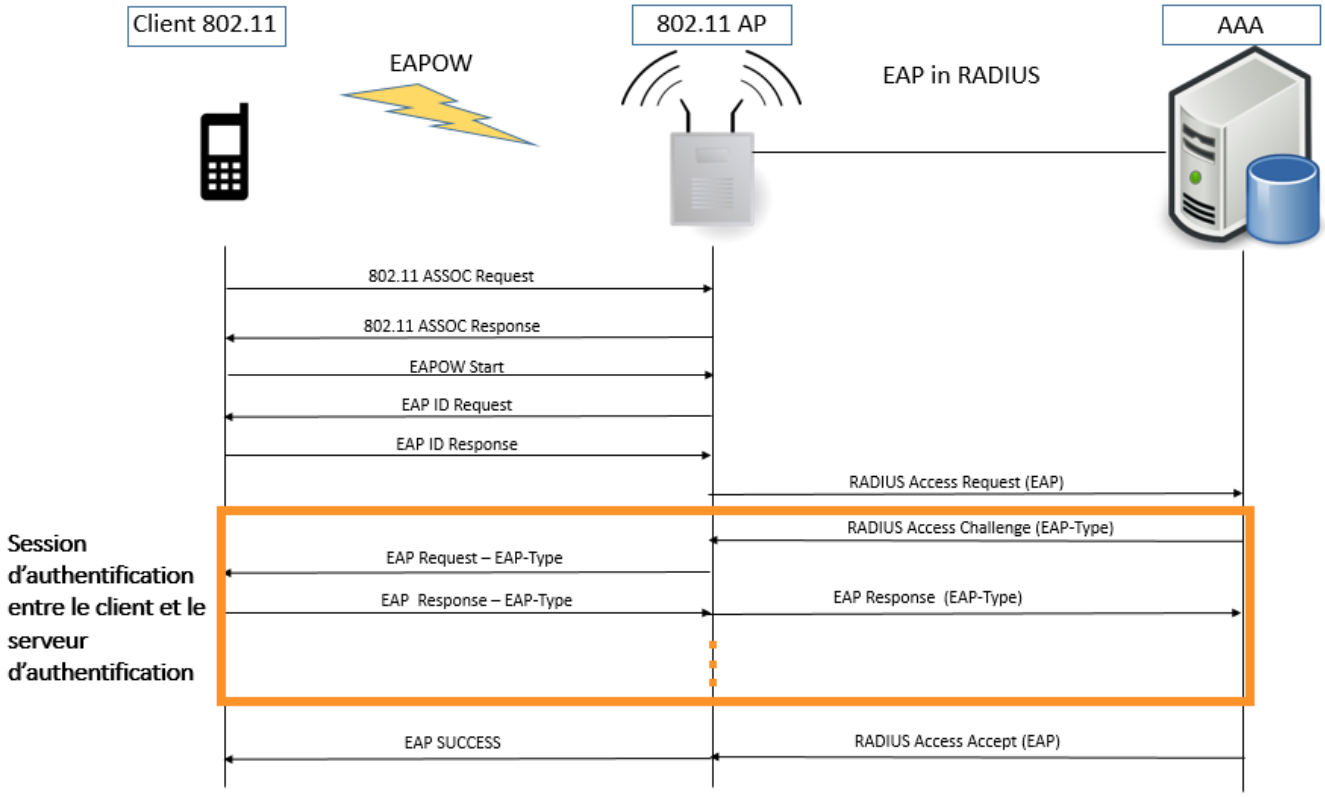

Figure 13 : Authentification 802.1X-EAP

Les messages échangés lors de la session d'authentification entre le client et le serveur RADIUS varient en fonction de la méthode EAP utilisée. Dans la Figure 14 nous détaillons l'échange fait entre le client et le serveur RADIUS pour l'EAP-TLS. Cet échange vient en complément de la Figure 13 pour l'échange de messages lors de l'authentification.

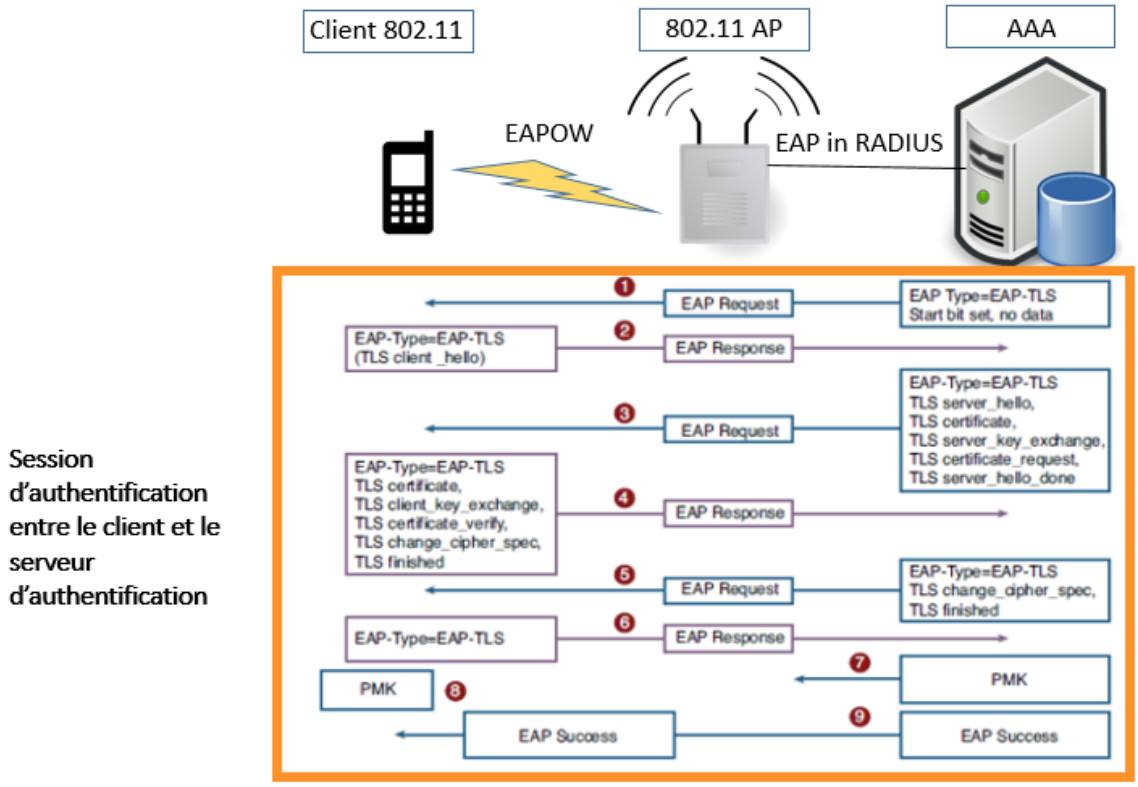

Figure 14 : Session d'authentification EAP-TLS

Pour pouvoir mettre en place ce système d'authentification au niveau d'un point d'accès Wi-Fi, il est nécessaire d'utiliser une autorité d'authentification (Authentication, Authorization, Accounting ou AAA) tel qu'un serveur RADIUS.

2.2.2.4 RADIUS

Comme décrit précédemment, le serveur d'authentification est un acteur de base du 802.1X et donc constitue la pierre angulaire de la norme Hotspot2.0. La description rapide que nous allons en faire nous permettra de mieux comprendre par la suite comment virtualiser un tel équipement.

RADIUS (Remote Authentication Dial-In User Service), est un protocole développé par l'entreprise Livingston dans le début des années 1991 et standardisé par la suite par l'IETF. Il permet une authentification standard, défini par un certain nombre de RFC dont la plus importante est la RFC 2865 [59]. RADIUS est basé sur un système de client/serveur. Dans ce système, le client cherche à s'authentifier auprès du serveur pour pouvoir accéder aux ressources du réseau. C'est un protocole largement utilisé chez les FAI (Fournisseurs d'Accès à Internet) car il est standard et propose entre autres des fonctionnalités de billing permettant de facturer avec précision les clients.

RADIUS se compose d'un serveur (le serveur RADIUS), rattaché à une base d'identification (une simple base de données, un annuaire LDAP, etc.) et un ou plusieurs client(s) RADIUS, appelé NAS (Network Access Server), qui servent à relayer la communication entre le client final et le serveur. Dans notre cas, le NAS sera le point d'accès Wi-Fi (car il sert d'intermédiaire, il est proprement appelé client RADIUS), et le client final est le périphérique mobile cherchant à se connecter au point d'accès Wi-Fi. L'ensemble des échanges entre le NAS et le serveur RADIUS sont chiffrées et authentifiées grâce à un mot de passe.

Le protocole RADIUS fonctionne comme suit:

- Un équipement mobile envoie une requête au point d'accès Wi-Fi afin de pouvoir s'associer ;
- Le point d'accès Wi-Fi relaye la requête au serveur RADIUS ;
- Le serveur RADIUS recherche dans sa base de données d'identification afin de vérifier si le client est légitime à accéder au réseau à travers le point d'accès Wi-Fi ou non, quatre réponses sont possibles :
    - ACCEPT : l'identification a réussi car le client a été antérieurement créé dans la base de données reliée au serveur ;
    - REJECT : l'identification a échoué car le client n'a pas été antérieurement créé dans la base de données;
    - CHALLENGE : le serveur RADIUS désire collecter des informations supplémentaires sur l'utilisateur final.
    - Il existe une réponse appelée CHANGE PASSWORD où le serveur RADIUS demande à l'utilisateur un nouveau mot de passe.

Cette réponse est un nouveau type de réponse introduit par Microsoft et ne figure pas dans la RFC 2865 de RADIUS. Ce type de réponse est décrit dans MS-CHAPv2 (RFC 2759) et est implémenté dans tous les serveurs RADIUS même s'il ne fait pas parti originalement du protocole standard.

Une fois l'authentification réussie, Le serveur RADIUS autorise le client final à accéder aux services du réseau précédemment inaccessibles. La Figure 15 illustre l'architecture tripartite constituée du client mobile 802.11, du point d'accès Wi-Fi (client RADIUS) et le serveur RADIUS. Le lien en pointillé rouge représente schématiquement l'accès aux services que demande le client mobile. Cette liaison ne pourra être établie que si l'authentification se fait avec succès (flèches pointillées noires).

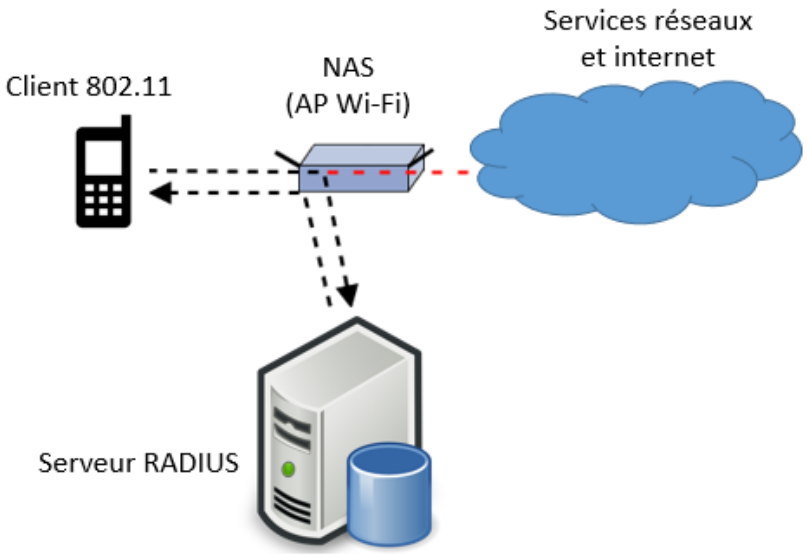

Figure 15 : Authentification avec un serveur RADIUS

2.2.3 L'architecture Hotspot2.0

Nous avons ainsi vu les différents protocoles clés de Hotspot2.0 : IEEE 802.11u, IEEE 802.1X, les méthodes EAP (à chaque méthode correspond un outil d'accès, « credential » en anglais), et l'IEEE 802.11i ; Les trois derniers cités font

partis de la certification WPA2-Enterprise de la Wi-Fi Alliance, et sont standardisés sur tous les smartphones. WPA2-Enterprise est l'un des niveaux de sécurité le plus élevé dans les réseaux Wi-Fi, il permet de sécuriser tout le processus d'échange d'informations entre un client et un point d'accès Wi-Fi. Le résultat final est un processus qui est tout aussi sécurisé et facile à utiliser que ce qui existe dans le monde cellulaire.

Le protocole IEEE 802.11u quant à lui permet à un appareil mobile d'avoir un dialogue avec le point d'accès Wi-Fi (pré-association) afin de déterminer les capacités que le réseau peut supporter, avant même de s'y connecter. Les deux protocoles que 802.11u utilise pour y arriver sont le Generic Advertisement Service (GAS) et le Access Network Query Protocol (ANQP). Ces protocoles fonctionnent au-dessus de 802.11 et permettent le bon fonctionnement de Hotspot2.0. La Figure 16 illustre ce qu'on vient d'expliquer avec l'empilement protocolaire qui y correspond.

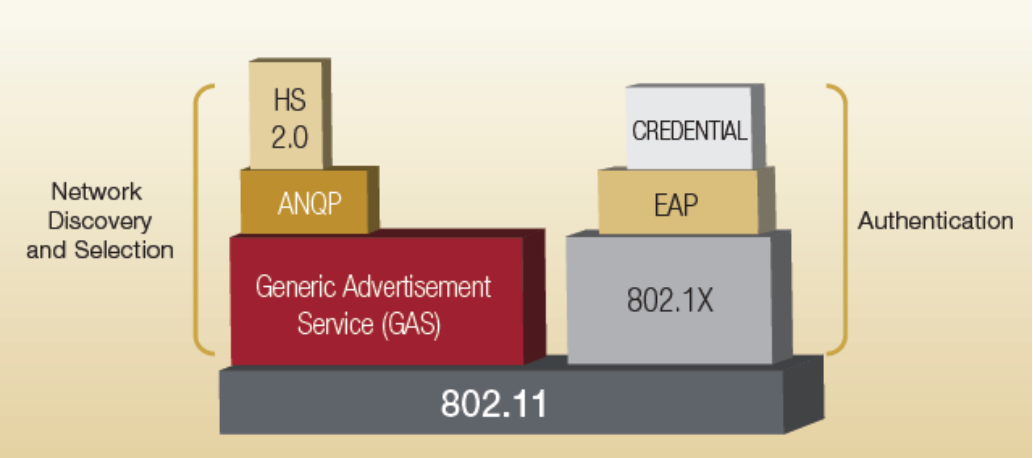

Figure 16 : Les composants de Hotspot2.0

La Figure 17 illustre un client qui utilise son smartphone pour accéder à un service à travers le réseau cellulaire de son opérateur. Dans sa mobilité, le client s'est retrouvé dans une zone de couverture d'un point d'accès Wi-Fi certifié Passpoint appartenant à son opérateur. Grace aux trames GAS/ANQP, le point d'accès Wi-Fi annonce qu'il est certifié Passpoint et qu'il peut assurer la continuité de service au client sur son réseau Wi-Fi. Le client, à travers son interface Wi-Fi et son logiciel de scanning, analyse les informations et initie

l'association avec l'AP. Avant de pouvoir s'associer, le client doit s'authentifier à travers le protocole 802.1X. Le point d'accès Wi-Fi agit alors en tant que NAS transparent et transmet la demande de connexion au serveur d'authentification RADIUS. Le serveur RADIUS tente d'authentifier le client en vérifiant son identité (certificat X.509, une identité SIM, etc.). Si le client mobile est légitime, le serveur RADIUS permet au client d'accéder aux services à travers ce point d'accès Wi-Fi certifié Passpoint.

Il en résulte que le client libère une ressource radio de son réseau cellulaire, et continue d'accéder à ses services de façon transparente sur un réseau Wi-Fi sans aucune intervention humaine.

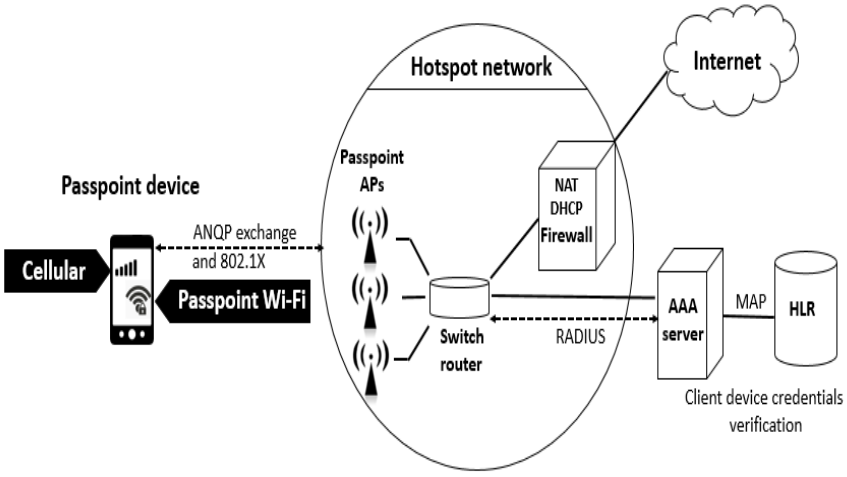

Figure 17 : Architecture Hotspot2.0

2.2.4 La virtualisation de points d'accès Wi-Fi Hotspot2.0

Le but de notre travail est de retranscrire toute la norme Wi-Fi Hotspot2.0 dans un environnement virtualisé comme décrit dans le Chapitre 1. Cela nécessitera dans un premier temps d'introduire dans notre architecture de point d'accès Wi-Fi virtuel, un serveur d'authentification pour obtenir dans une première étape un point d'accès Wi-Fi virtuel de niveau de sécurité WPA2-Enterprise. Les points d'accès Wi-Fi virtuels sécurisés que nous avons développé durant notre travail ont donné lieu à une démonstration durant un workshop [9].

Une fois cette sécurité atteinte, il nous faudra développer aussi bien du côté client que du côté point d'accès Wi-Fi les nouvelles fonctionnalités

introduites par Hotspot2.0 dont notamment la pré-sélection et découverte de réseau 802.11u.

Comme indiqué précédemment, 802.1X est un protocole de machine à machine dans le sens où il nécessite deux machines pour transférer des données binaires sur un protocole de réseau. Du côté client, un logiciel appelé *wpa_supplicant* doit être déployé. Du côté serveur d'authentification, le serveur RADIUS doit être configuré minutieusement pour permettre à l'AP (jouant le rôle NAS) de s'y connecter et aux clients mobiles d'être authentifiés. Et finalement du côté AP, une configuration assez pointue au niveau du *hostapd* dont nous avons parlé dans le 1er Chapitre doit être établie pour permettre de relayer les informations d'authentifications entre le client et le serveur RADIUS, et les trames GAS/ANQP entre le client mobile et l'AP.

2.2.4.1 L'environnement logiciel

L'environnement logiciel se base sur celui déjà décrit dans la sous-section 1.5.4.2 du Chapitre 1 pour la création des points d'accès Wi-Fi virtuels avec néanmoins une recompilation du hostapd après quelques modifications que nous expliquerons par la suite. Le nouvel élément introduit dans l'architecture est le logiciel Freeradius pour nous permettre d'ajouter l'authentification 802.1X dans notre système.

Freeradius est le serveur RADIUS open source le plus populaire et le plus largement déployé dans le monde. Il fournit l'authentification, l'autorisation, et la traçabilité (Authentication, Authorization, Accounting : AAA). Il est aussi largement utilisé par la communauté universitaire (comme eduroam, qui est le service d'itinérance pour l'accès à internet mobile, développé pour le milieu de la recherche et de l'éducation à l'international).

Freeradius a débuté en Août 1999 avec Alan DeKok et Miquel van Smoorenburg. Miquel avait précédemment écrit le logiciel Cistron RADIUS, qui a été largement adoptée lorsque le Serveur Livingston n'était plus en service. Freeradius a été développé en utilisant une conception modulaire pour encourager la participation de la communauté des développeurs.

La popularité de Freeradius peut être accordée à la multitude d'avantages supplémentaires qu'il offre, bien au-dessus et au-delà de celles de la grande variété

d'autres serveurs RADIUS. Freeradius est basée sur des fonctionnalités riches, modulaires, et évolutives, qui offrent les avantages suivants à réseau administrateurs:

- **Fonctionnalités**

Plus de types d'authentification sont supportées par Freeradius que par tout autre serveur open source. Par exemple, Freeradius a été le premier serveur RADIUS open source à supporter les méthodes EAP. Freeradius est le seul serveur RADIUS open source à soutenir les serveurs virtuels, et qui dans notre cas a été l'élément décisif de choix. L'utilisation de serveurs virtuels signifie que les implémentations complexes sont simplifiées et que les coûts de support et de maintenance sont considérablement réduits.

- **Modularité**

La conception modulaire du protocole permet à Freeradius d'être facile à comprendre. L'interface modulaire simplifie également l'ajout ou la suppression de modules (par exemple, si une fonction n'est pas nécessaire pour une configuration particulière, le module peut être facilement retiré). Une fois que le module est retiré, il n'affecte pas les performances du serveur, l'utilisation de la mémoire, ou de la sécurité. Cette flexibilité permet au serveur de fonctionner sur des plates-formes allant des systèmes embarqués aux machines multi-core avec des GB de RAM.

- **Évolutivité**

Un unique serveur RADIUS peut facilement passer du traitement d'une requête par seconde à la manipulation de milliers de demandes par seconde, tout simplement en reconfigurant quelques paramètres par défaut. Beaucoup de grandes organisations (millions de clients) dépendent de Freeradius pour leurs besoins de AAA. Souvent, un seul serveur Freeradius suffit à combler les besoins de ces grandes organisations.

2.2.4.2 Implémentation du côté serveur d'authentification

Pour notre solution nous avons choisis de mettre en place Freeradius comme serveur d'authentification implémentant l'EAP-TLS qui est considérée comme l'une des méthodes d'authentification la plus sécurisée (authentification mutuelle entre le client et le serveur RADIUS par le biais de certificats côté client et côté serveur) mais aussi comme la plus complexe à mettre en place. L'EAP-TLS se veut une méthode d'authentification universelle car elle ne nécessite pas forcément de carte SIM et peut être mis en place dans tout objet connecté (caméra IP, imprimante, smart TV, etc.).

Comme point de départ nous créons une machine virtuelle Linux sous Xen dans laquelle nous commencerons par installer le module OpenSSL nécessaire à la compilation des différentes applications qui suivront. OpenSSL est un outil de chiffrement comprenant de nombreuses librairies cryptographiques utilisées pour la gestion de la couche TLS. La version utilisée est la 1.0.1p et est non concernée par la faille Heartbleed [10] qui avait été découverte en 2014 et qui touchait les versions d'OpenSSL de la 1.0.1 à la 1.0.1f (il était possible de dérober les clés secrètes des certificats X.509 donnant accès à toutes les informations confidentielles de l'utilisateur). Une fois OpenSSL compilé sans erreurs (il arrive que certaine librairies supplémentaires soient requises en fonction de la distribution utilisée), on doit le configurer à travers son fichier *openssl.cnf* qu'on trouve sous */usr/local/openssl-certgen/ssl.* Il s'agit d'un formulaire d'informations sur l'autorité de certification à compléter avec différents champs tels que le pays, la ville, l'organisation d'appartenance, le nom, une adresse mail, etc. Une fois le fichier configuré, l'installation d'OpenSSL est terminée, il nous faut à présent créer les différents certificats.

Le premier certificat à générer est le certificat root qui sera l'autorité de certification qui servira par la suite à signer les autres certificats générés. Il est à noter que dans le cadre de l'EAP-TLS chaque machine devra avoir à sa disposition son propre certificat personnel et le certificat root. Pour créer le certificat root, il suffit de lancer sous */root/certs* le script *./CA.root* fourni par OpenSSL. Le script va générer les certificats *root.pem, root.p12* et *root.der.* L'extension pem est le format le plus courant échangé avec une autorité d'authentification, et à l'avantage de pouvoir contenir plusieurs certificats et clés

dans un seul et même fichier. L'extension der est la forme binaire de la version ASCII du pem et est très utilisée sur les plateformes java. L'extension p12 est surtout utilisée sous Windows, c'est un format pouvant contenir les certificats intermédiaires et des clés de chiffrement. Il est à noter que des lignes de commandes OpenSSL nous permettent de passer d'un format à un autre. Dans notre cas, le certificat root.pem sera utilisé au niveau de Freeradius, et root.der sera distribué au client.

Une fois l'autorité de certification créée, nous devons créer le certificat serveur qui sera installé au niveau du Freeradius. Il nous faut lancer un autre script fourni par OpenSSL : *./CA.svr *Common Name** avec en argument un « Common Name », qui sera le nom du serveur Freeradius, nous choisirons « serveur ». Nous obtenons ainsi les certificats *serveur.pem, serveur.p12* et *serveur.der.*

Et finalement, nous créons le certificat du client mobile avec le script *./CA.clt *Common Name** avec l'argument « Common Name » le nom du client. Il est important de choisir un nom unique pour chaque client, car contrairement aux certificats root et serveur (à moins de faire appel à un cluster de Freeradius, dans ce cas-là nous nous trouverons face à plusieurs certificats serveur), les certificats clients sont multiples et doivent désigner de façon exclusive un utilisateur. Nous prenons pour exemple « client1 » comme Common Name. Nous obtenons *client1.pem*, *client1.p12* et *client1.der*.

A ce stade, nous avons créé tous les certificats dont nous avons besoin au sein de la machine virtuelle. Nous pouvons passer à l'installation et à la configuration du serveur Freeradius. La version de Freeradius utilisée est la 2.2.0. La compilation de celui-ci requiert de nombreuses bibliothèques pour éviter que des erreurs surviennent pendant la compilation. Nous devons installer les modules *libssl-dev, snmp* et *libltdl3-dev.* Une fois que ces modules sont installés au niveau de notre machine virtuelle, nous pouvons lancer l'installation de Freeradius avec une configuration spécifique pour la compilation pour spécifier nos besoins ; On spécifiera lors de la configuration le paramètre *–sysconfdir=/etc* pour pouvoir retrouver tous nos fichiers de configuration sous le répertoire */etc/raddb*. Dès lors que la configuration se passe sans aucun message d'erreur (l'arbre des dépendances des modules de Freeradius est très dense, il n'est pas rare que lors de la configuration des librairies supplémentaires soient demandées) on peut lancer la compilation et l'installation.

Pour résumer, nous avons maintenant à notre disposition le serveur Freeradius prêt à être utilisé, et les certificats que nous avons créés grâce à OpenSSL. Il faut maintenant peupler le serveur Freeradius des différents certificats nécessaires à son fonctionnement. On se rend dans le dossier */etc/raddb* et nous allons effacer les certificats présents par défaut, et copier nos certificats root et serveur dans le dossier */etc/raddb/certs*. Lors de la première utilisation du serveur Freeradius, il nous faut créer deux fichiers dh et random qui sont tous les deux des fichiers contenant des informations aléatoires nécessaires au fonctionnement de TLS dans le module EAP, une simple commande *date* dans chacun des fichiers suffit. Le serveur est maintenant peuplé de certificats et contient les fichiers nécessaires, il faut à présent le configurer pour activer l'EAP-TLS. Il y a de nombreux fichiers de configuration dans le Freeradius qui rentrent en jeu pour son bon fonctionnement. C'est d'ailleurs ce qui le rend difficile à manipuler. Il nous faut modifier la configuration de la méthode EAP utilisée et les certificats utilisés dans *eap.conf,* modifier la configuration du point d'accès Wi-Fi à contacter (NAS) par Freeradius dans *clients.conf,* et finalement modifier la configuration des clients autorisés dans *users.*

Fichier eap.conf

Il faut spécifier qu'EAP-TLS est la méthode à utiliser :

```
default_eap_type = tls
```

Il suffit par la suite de retrouver le module tls dans le fichier et décommenter la partie dont nous avons besoin et changer les chemins des certificats utilisés (différents de ceux qui étaient présents par défaut).

```
tls {
      private_key_password = whatever
      private_key_file = ${raddbdir}/certs/serveur.pem

      certificate_file = ${raddbdir}/certs/serveur.pem
      CA_file = ${raddbdir}/certs/root.pem

      dh_file = ${raddbdir}/certs/dh
      random_file = ${raddbdir}/certs/random
      fragment_size = 1024

      include_length = yes

      #check_crl = yes

      check_cert_cn = %{User-Name}
      }
}
```

Fichier clients.conf

Ce fichier va permettre au serveur Freeradius de connaitre la liste des points d'accès Wi-Fi virtuels qui sont autorisés à communiquer avec lui. Freeradius et le point d'accès Wi-Fi virtuel partagent une clé de chiffrement pour l'échange de données. Par défaut le fichier autorise le 127.0.0.1 (localhost) à communiquer avec le serveur avec la clé « testing123 ». A ce stade, il faut connaitre l'adresse IP du point d'accès Wi-Fi virtuel que nous allons utiliser, prenons pour exemple 10.0.0.20, l'édition du fichier va consister à rajouter ces lignes :

```
Client 10.0.0.20 {
Secret    = Virtuor_security
Shortname = Virtuor
Nastype   = other
}
```

Fichier users

Dans ce fichier nous devons spécifier la liste des utilisateurs qui se connectent au serveur RADIUS (les points d'accès Wi-Fi) avec la méthode EAP correspondante. Pour forcer une méthode EAP spécifique (dans notre cas EAP-TLS) il faut le mentionner à travers ce jeu de paramètres dans le fichier :

```
`client` Auth-Type := EAP,   EAP-Type :=  EAP-TLS
```

Il faut rajouter autant de lignes de déclaration de clients que de NAS utilisés, pour permettre au serveur Freeradius de gérer plusieurs points d'accès en même temps.

Comme dernier point important, il faut doter le serveur RADIUS d'une adresse IP pour permettre la communication sur le réseau, dans notre expérimentation nous lui affectons l'adresse 10.0.0.30. A travers la configuration établie, la machine virtuelle que nous avons créée se voit dotée de toutes les fonctionnalités d'un serveur RADIUS physique. Le serveur Freeradius est prêt à

l'emploi, il faut alors configurer le point d'accès Wi-Fi pour permettre la communication RADIUS entre les deux entités.

2.2.4.3 La modification de hostapd

Pour être en mesure d'activer Hotspot2.0 dans le point d'accès Wi-Fi virtuel, nous avons dû recompiler le daemon hostapd qui par défaut ne supporte pas les nouvelles fonctionnalités ajoutées par la norme. Dans le fichier caché *.config* présent dans l'arborescence */etc/hostapd* au niveau du Dom0 nous activons les fonctionnalités du protocole IEEE802.11u en ajoutant les deux paramètres suivants :

```
CONFIG_HS20=y
CONFIG_INTERWORKING=y
```

Une fois ces modifications faites, le daemon doit être recompilé et réinstallé de nouveau pour permettre l'intégration des nouvelles fonctionnalités. Le hostapd est maintenant prêt pour supporter le 802.11u que nous devons configurer dans le fichier de configuration *hostapd.conf*, c'est aussi dans ce fichier de configuration que nous devons mentionner les paramètres du serveur RADIUS avec lequel le point d'accès doit communiquer.

```
ssid="Hotspot2.0"
[...]
# Activation de 802.11u
interworking=1
hs20=1
access_network_type=14
venue_group=2
venue_type=0
roaming_consortium=2233445566
venue_name=eng:Place Jussieu
# Activation de WPA2-Enterprise
ieee8021x=1
```

```
auth_algs=1
wpa=2
wpa_key_mgmt=WPA-EAP
wpa_pairwise=CCMP
rsn_pairwise=CCMP
own_ip_addr=10.0.0.20
auth_server_addr=10.0.0.30
auth_server_port=1812
auth_server_shared_secret=Virtuor_security
acct_server_addr=10.0.0.30
acct_server_port=1813
acct_server_shared_secret=Virtuor_security
```

La configuration de hostapd.conf constitue la carte d'identité du point d'accès Wi-Fi. La configuration donnée plus haut est divisée en deux parties : la première pour l'activation du protocole 802.11u avec toutes les données que doivent comporter les trames GAS/ANQP, et la seconde pour activer la sécurité WPA2-Enterprise par l'ajout du serveur RADIUS pour activer le 802.1X. Cette configuration vient s'ajouter à la configuration de base d'un point d'accès Wi-Fi standard (SSID, canal, driver, débits utilisés, etc.).

Les différents champs utilisés pour l'activation de Hotspot2.0 se traduisent comme suit :

- interworking : permet d'activer IEEE 802.11u
- HS20 : permet le support de Hotspot2.0
- access_network_type : permet de spécifier le type de réseau visité (dans notre cas :14, qui signifie réseau de tests et d'expérimentations)
- venue_group et venue_type permettent de spécifier la nature de l'endroit visité, ces deux champs sont spécifiés dans le standard IEEE 802.11u-2011, la combinaison des deux permet de mentionner au client mobile qu'il s'agit d'un réseau Wi-Fi appartenant à un café par exemple (venue_group=7, venue_type=1). Dans notre cas (group=2, type=0) signifie qu'il s'agit d'une entreprise non spécifiée.
- roaming_consortium : Il s'agit d'un élément clé du hotspot2.0, il permet de spécifier quels sont les partenaires d'itinérance du fournisseur d'accès

de ce point d'accès. Chaque numéro de roaming_consortium désigne de manière unique un fournisseur d'accès. Nous pouvons rajouter autant de champs roaming_consortium que de partenaires d'itinérance. Ce champ permet ainsi à l'utilisateur abonné au fournisseur de numéro de roaming_consortium 22446688 de pouvoir se connecter à ce point d'accès. Si plusieurs roaming_consortium sont supporté par le point d'accès, ils seront tous mentionner dans la liste d'informations de la trame ANQP.

- venue_name : ce sont des informations supplémentaires sur le lieu où se situe le point d'accès Wi-Fi (metadata). Le nom du fournisseur, de l'institution ou même l'adresse et le numéro de téléphone peuvent être fourni dans ce champs supplémentaire permis par ANQP.

D'autres nombreux champs introduits par 802.11u peuvent être ajouté en fonction des informations que le fournisseur du point d'accès Wi-Fi veut diffuser. Nous nous contentons d'une configuration minimaliste pour prouver le fonctionnement de notre point d'accès Wi-Fi virtuel Passpoint. Les autres champs utilisés dans hostapd.conf servent à définir les règles de sécurité et communication entre le point d'accès Wi-Fi et le serveur Freeradius déjà crée :

- ieee8021x : permet d'activer l'authentification IEEE 802.1X sur le point d'accès Wi-Fi.
- auth_algs : spécifie l'algorithme d'authentification. En Wi-Fi en distingue l'authentification ouverte et l'authentification à clé partagée. Dans le cas de 802.1X l'authentification ouverte est nécessaire (auth_algs=1).
- wpa : permet d'activer IEEE 802.11i/RSN pour l'authentification 802.1X et le chiffrement AES.
- wpa_key_mgmt : permet de spécifier que l'authentification se fait à travers une méthode EAP (requis par 802.1X).
- wpa_pairwise et rsn_pairwise : permettent d'activer le chiffrement AES en mode counter avec CBC-MAC spécifié par la RFC 3610 [60] (requis par 802.11i).

- own_ip_addr : il s'agit de l'adresse du point d'accès Wi-Fi marchant en tant que NAS entre le client mobile et le serveur RADIUS. Ce champs ne substitut pas la configuration de l'adresse IP du point d'accès.
- auth_server_addr : adresse IP du serveur RADIUS pour la fonction d'authentification
- auth_server_port : il s'agit du numéro de port UDP pour l'authentification avec un serveur RADIUS.
- auth_server_shared_secret : secret partagé entre le NAS et le serveur RADIUS pour la fonction d'authentification.
- acct_server_addr : adresse IP du serveur RADIUS pour la fonction d'accounting.
- acct_server_port : il s'agit du numéro de port UDP pour l'accounting avec un serveur RADIUS.
- acct_server_shared_secret : secret partagé entre le NAS et le serveur RADIUS pour la fonction d'accounting.

A ce stade de l'expérimentation, le serveur Freeradius est prêt et l'interface Wi-Fi virtuelle a été configurée pour supporter Hotspot2.0. Une machine virtuelle assurant les fonctions de routages, de DHCP et de résolution de nom de domaines (la même qu'utilisée dans le Chapitre 1 en section 1.5.4) doit être mise en marche pour être rattachée à l'interface Wi-Fi virtuelle. Cette machine comme mentionné plus haut, devra avoir pour adresse IP 10.0.0.20 et devra être bridgée avec le serveur Freeradius d'adresse 10.0.0.30 comme illustré dans la Figure 18.
La topologie de la solution que nous proposons peut aussi bien servir à instancier un point d'accès Wi-Fi Hotspot2.0 virtuel qu'un point d'accès Wi-Fi virtuel sécurisé WPA2-Enterprise (dans lequel cas, nous devons retirer la configuration 802.11u du hostapd.conf).

Le serveur Freeradius doit être lancé en premier avec la commande *radiusd –X* qui doit être lancée dans la machine virtuelle Freeradius. L'interface Wi-Fi virtuelle est lancée par la suite avec le daemon hostapd auquel nous spécifions le fichier de configuration hostapd.conf que nous avons édité. Le point d'accès Wi-Fi Hotspot2.0 virtuel est désormais en marche, il faut à présent développer un client capable de se connecter à ce point d'accès Wi-Fi Hotspot2.0.

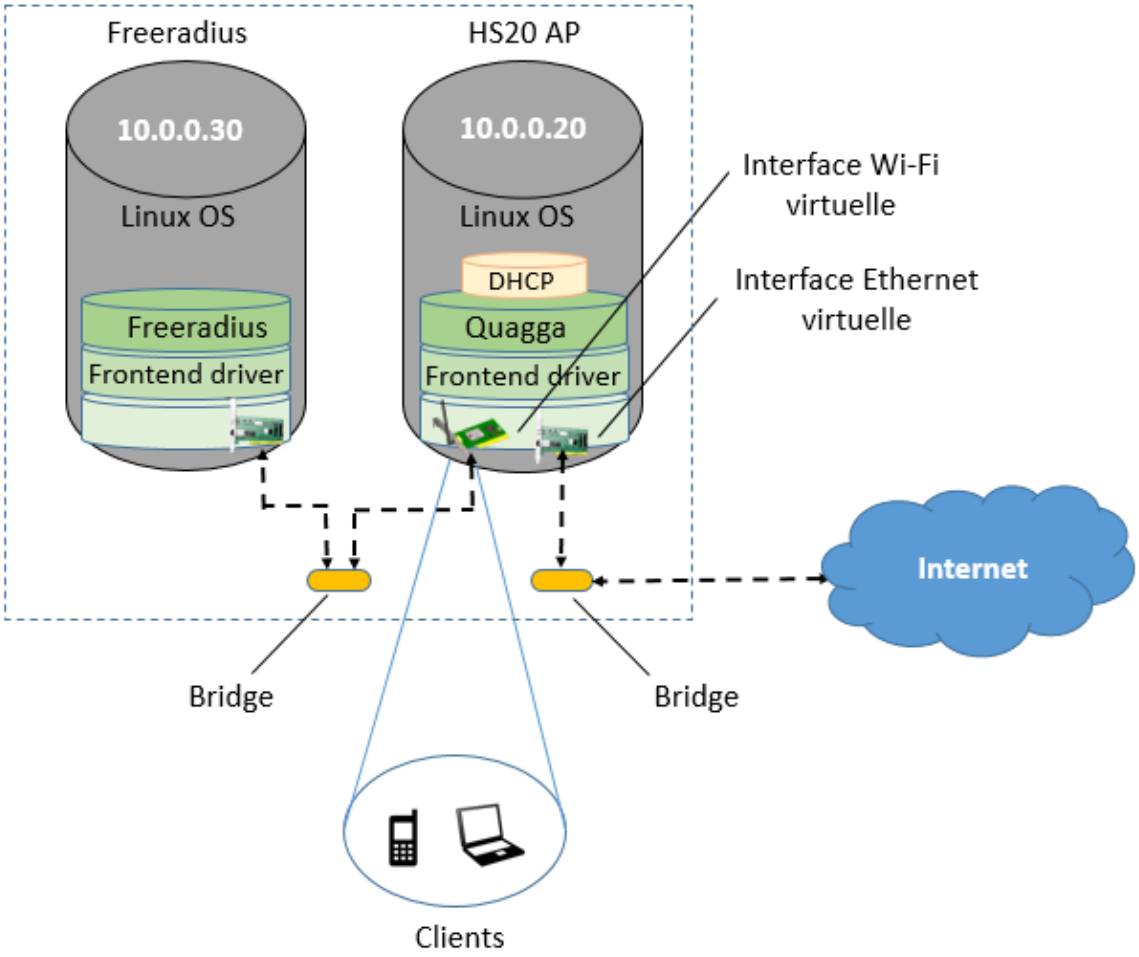

Figure 18 : Architecture de notre point d'accès Wi-Fi Hotspot2.0 virtuel

2.2.4.4 La configuration du côté client

La configuration du client se fait à peu près sur le même principe de base que le point d'accès à travers l'outil wpa_supplicant. Le wpa_supplicant est un client Wi-Fi multi-plateforme Linux, BSD, Mac OS X et Windows, qui supporte le WEP, WPA et WPA2 (IEEE 802.11i / RSN : Robust Secure Network). Cet outil peut être utilisé sur les ordinateurs fixes, ordinateurs portables ou systèmes embarqués. Il s'agit d'un daemon (il s'exécute en arrière-plan) qui se charge de la négociation des clés avec les autorités d'authentification (serveur RADIUS), il contrôle le roaming dans les réseaux Wi-Fi et également l'authentification et l'association du driver Wi-Fi de l'équipement mobile. Son fonctionnement de base est décrit dans les étapes suivantes :

- wpa_supplicant effectue des appels système au kernel driver pour balayer les réseaux Wi-Fi voisins à portée de l'antenne Wi-Fi du dispositif.
- wpa_supplicant sélectionne un SSID en fonction de la configuration déjà préenregistrée, tel qu'un réseau déjà visité par exemple.
- wpa_supplicant demande au kernel driver de s'associer au BSS choisis.
- S'il s'agit d'une connexion WPA-EAP (ce qui est notre cas), le module 802.1X intégré dans le wpa_supplicant complète l'authentification EAP avec le serveur d'authentification (qui se trouve derrière le point d'accès Wi-Fi).
- La PMK (Pairwise Master Key) est ensuite reçue par le module 802.1X du wpa_supplicant.
- wpa_supplicant complète par la suite le 4-way handshake avec l'AP, pour dériver et échanger les clés unicast/multicast à partir du PMK avant que la connexion ne soit établie.

Le wpa_supplicant se base sur un fichier de configuration wpa_supplicant.conf dans lequel des profils de réseaux Wi-Fi préconfigurés ou déjà visités sont enregistrés. Quand le driver Wi-Fi balaye les SSID disponibles, wpa_supplicant compare ceux-ci aux réseaux présents dans le fichier de configuration. Si une correspondance est trouvée, wpa_supplicant lance le processus d'association au point d'accès Wi-Fi correspondant. Si plusieurs AP Wi-Fi correspondent aux profils préconfigurés, un champ de priorisation définira quel point d'accès doit être sélectionné.

Le client utilisé pour nos expérimentation est un PC portable Asus UL20A, avec un processeur Intel Core 2 Duo SU7300, 2GB de RAM, disque dur 5400 RPM SATA II, et une carte Wi-Fi Atheros AR5BXB72. Le système d'exploitation installé sur notre machine client mobile est Ubuntu 12.04 LTS, qui est une version très stable de la distribution. Pour pouvoir concevoir proprement notre client Wi-Fi, il faut dans un premier temps désactiver l'utilitaire de connexion Wi-Fi présent par défaut dans le système à travers la commande :

```
sudo service network-manager stop
```

L'icône du Wi-Fi donnant l'état de la connexion du PC ainsi que tous les points d'accès Wi-Fi disponibles disparait. Nous téléchargeons la version 2.0 de

wpa_supplicant que nous devons modifier avant compilation. Pour être en mesure d'activer Hotspot2.0 dans le client Wi-Fi, nous devons activer les nouvelles fonctionnalités ajoutées par la norme. Dans le fichier *.config* présent dans l'arborescence */etc/wpa_supplicant* nous activons les fonctionnalités du protocole IEEE802.11u en ajoutant les deux paramètres suivants :

```
CONFIG_HS20=y
CONFIG_INTERWORKING=y
```

Il est à noter que le package OpenSSL doit être indispensablement installé sur le client pour le bon fonctionnement du module 802.1X de wpa_supplicant. Une fois ces modifications faites, le wpa_supplicant doit être compilé et installé pour permettre l'intégration des nouvelles fonctionnalités. Le wpa_supplicant est maintenant prêt pour supporter le 802.11u que nous devons configurer dans le fichier de configuration *wpa_supplicant.conf.*

Chaque réseau Wi-Fi est balisé dans le fichier de configuration *wpa_supplicant.conf* par :

```
network={
    ...
}
```

C'est entre ces accolades que la description du point d'accès Wi-Fi auquel on désire se connecter, doit être faite pour pouvoir initier une association. Nous devons spécifier le SSID, le type d'authentification, la méthode d'authentification, mentionner qu'il s'agit d'un point d'accès Wi-Fi Hotspot2.0, donner le chemin d'accès aux certificats clients et root pour pouvoir s'authentifier avec de l'EAP-TLS auprès du serveur Freeradius, etc. La configuration du client permettant de se connecter au point d'accès Wi-Fi Hotspot2.0 virtuel que nous avons créé est la suivante :

```
hs20=1
interworking=1
```

```
network={
    ssid="Hotspot2.0"
    roaming_consortium=2233445566
    scan_ssid=1
    key_mgmt=WPA-EAP
    pairwise=CCMP
    group=CCMP
    eap=TLS
    identity="user@example.com"
    ca_cert="/etc/cert/root.pem"
    client_cert="/etc/cert/client1.pem"
    private_key="/etc/cert/client1.pem"
    priority=100
}
```

Les champs renseignés sont presque similaires à ceux présents dans le fichier hostapd.conf du point d'accès Wi-Fi créé. Les champs hs20 et interworking sont nécessaires pour que le client soit en mesure de comprendre les trames GAS/ANQP émises par le point d'accès Wi-Fi Hotspot2.0 virtuel créé. Les nouveaux éléments ajoutés concernent les éléments d'authentification du client. En effet, lors de la connexion au point d'accès, le module 801.X de wpa_supplicant doit donner l'identité du client mobile à travers le champ *identity*, qui est le même utilisé dans le serveur Freeradius lors de la création du certificat client. Les champs ca_cert, client_cert, private_key sont là pour indiquer au module 802.1X de wpa_supplicant, où chercher les certificats lors de l'authentification avec le serveur Freeradius lorsque celui-ci les demandera. Ces certificats ont été créés dans la machine virtuelle Freeradius. Nous avons dû les transférer sur l'équipement du client mobile grâce à une clé USB (nous reviendrons sur ce point dans la section 2.3), et nous les avons mis sous l'arborescence */etc/cert* du client. A ce stade, le client est prêt à s'associer au point d'accès Wi-Fi Hotspot2.0, il suffit de lancer le daemon et le laisser tourner en arrière-plan dans le client à travers la commande :

```
wpa_supplicant –B -i wlan0 –c /etc/wpa_supplicant.conf
```

Le client Hotspot2.0, le point d'accès Wi-Fi Hotspot2.0 et le serveur Freeradius virtuel sont prêts à l'emploi. Nous allons voir quels sont les résultats que nous obtenons et si l'architecture créée permet de fournir du Wi-Fi Hotspot2.0 virtualisé.

2.2.4.5 Résultats

La topologie qui a été mise en place peut se résumer par la Figure 19 qui décrit l'état du système à l'état initial avant authentification du client. Les composantes du réseau et les liens en rouge représentent les éléments inaccessibles au client non encore authentifiés. Comme nous l'avons déjà mentionné, un client non authentifié ne peut avoir accès aux services du réseau (DHCP, DNS, internet). Le lien reliant le serveur Freeradius au bridge reste non accessible au client mobile même après authentification, le seul client légitime ayant le droit de communiquer avec le serveur d'authentification est le point d'accès Wi-Fi virtuel.

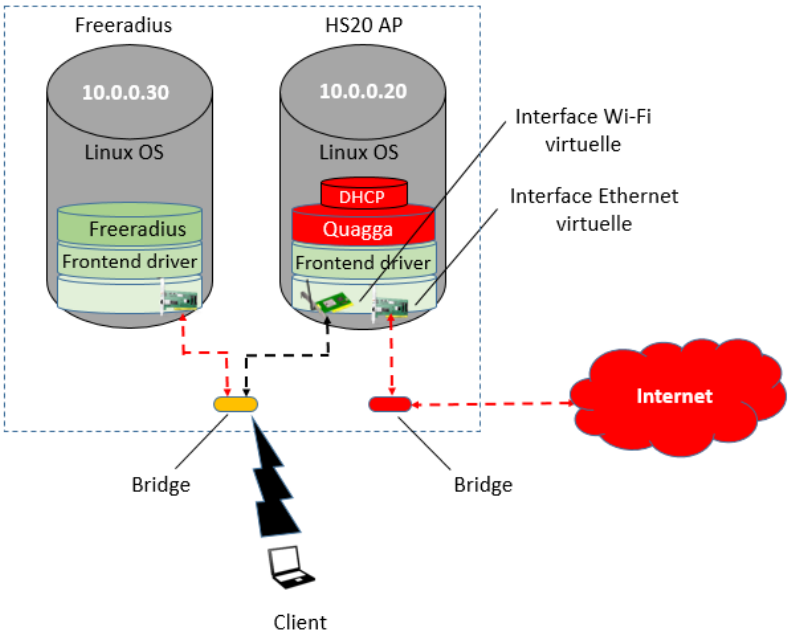

Figure 19 : Architecture Hotspot2.0 virtuel avec client non authentifié

Nous commençons par réaliser une capture du trafic dans l'air grâce à Wireshark installée sur une machine tiers. Wireshark est configuré en mode

promiscious pour sniffer tout le trafic du canal du réseau sans fil. Le but est de retrouver les trames 802.11 que le point d'accès Wi-Fi virtuel Hotspot2.0 que nous avons créé est entrain de diffuser.

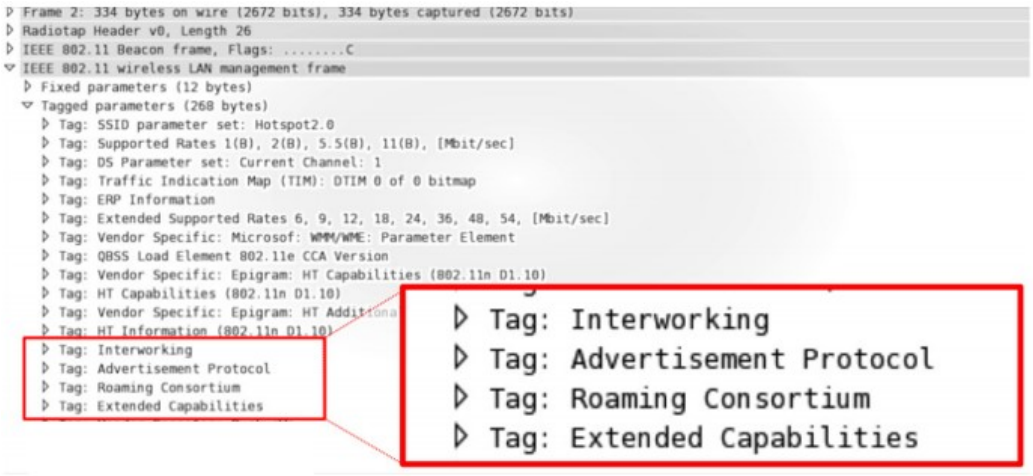

Figure 20 : Capture d'une trame Hotspot2.0

Nous constatons que de nouveaux IE (Information Element) sont présents dans la beacon frame (trame 802.11 annonçant la présence d'un AP et ses caractéristiques). Les nouvelles informations 802.11u rajoutées dans le *hostapd.conf* figurent bien dans la signalisation du point d'accès.

Pour comprendre ce qui se passe au niveau du client Hotspot2.0 tentant de se connecter à un point d'accès Wi-Fi virtuel Hotspot2.0, nous réalisons des captures de logs du wpa_supplicant au niveau du client :

```
<3>Starting ANQP fetch for 01:02:03:04:05:01
<3>RX-ANQP 01:02:03:04:05:01 ANQP Capability list
<3>RX-ANQP 01:02:03:04:05:01 Roaming Consortium list
<3>RX-HS20-ANQP 01:02:03:04:05:01 HS Capability List
<3>ANQP fetch completed
<3>INTERWORKING-AP 01:02:03:04:05:01 type=unknown
```

Ces messages nous indiquent que le wpa_supplicant a trouvé un AP qui supporte Hotspot2.0. Cet AP d'adresse MAC 01:02:03:04:05:01 à un profil qui correspond à l'un des profils préenregistré dans le fichier wpa_supplicant.conf, le client va alors tenter de s'y connecter :

```
<3>CTRL-EVENT-SCAN-RESULTS
<3>SME: Trying to authenticate with 01:02:03:04:05:01
(SSID='Hotspot2.0' freq=2412 MHz)
<3>Trying to associate with 01:02:03:04:05:01
(SSID='Hotspot2.0' freq=2412 MHz)
<3>Associated with 01:02:03:04:05:01
<3>CTRL-EVENT-EAP-STARTED EAP authentication started
<3>CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=13
<3>CTRL-EVENT-EAP-METHOD EAP vendor 0 method 13 (TLS)
selected
<3>CTRL-EVENT-EAP-SUCCESS EAP authentication completed
successfully
<3>WPA: Key negotiation completed with
01:02:03:04:05:01 [PTK=CCMP GTK=CCMP]
<3>CTRL-EVENT-CONNECTED - Connection to
01:02:03:04:05:01 completed (auth) [id=0 id_str=]
```

Nous observons dans cette deuxième partie du fichier de logs que le client s'associe à notre point d'accès Wi-Fi « Hotspot2.0 ». Le client ensuite se lance dans une phase d'authentification en utilisant EAP-TLS, s'authentifie avec succès, et procède à un échange de clés avec le point d'accès Wi-Fi. Le client Hotspot2.0 est connecté au point d'accès Wi-Fi virtuel Hotspot2.0.

Pour vérifier le statut du client, wpa_supplicant offre un mode interactif via l'outil *wpa_cli* :

```
> status
bssid=01:02:03:04:05:01
ssid=Hotspot2.0
id=0
mode=station
pairwise_cipher=CCMP
group_cipher=CCMP
key_mgmt=WPA2/IEEE 802.1X/EAP
wpa_state=COMPLETED
p2p_device_address=01:02:03:04:05:01
address=01:02:03:04:05:01
hs20=1
Supplicant PAE state=AUTHENTICATED
suppPortStatus=Authorized
EAP state=SUCCESS
selectedMethod=13 (EAP-TLS)
EAP TLS cipher=AES-128-SHA
```

Le statut de la connexion nous montre bien que le client est connecté à un point d'accès Wi-Fi Hotspot2.0, pour ce faire, le client s'est authentifié en EAP-

TLS. Cela démontre que notre solution développée est fonctionnelle et prouve le concept de virtualisation d'une architecture Hotspot2.0. Le système après authentification du client peut s'illustrer à travers la Figure 21. Le client après connexion a bien obtenu une adresse IP à partir du serveur DHCP et a pleinement accès au service internet. Seul le serveur Freeradius lui reste inaccessible (accessible seulement au NAS : l'AP).

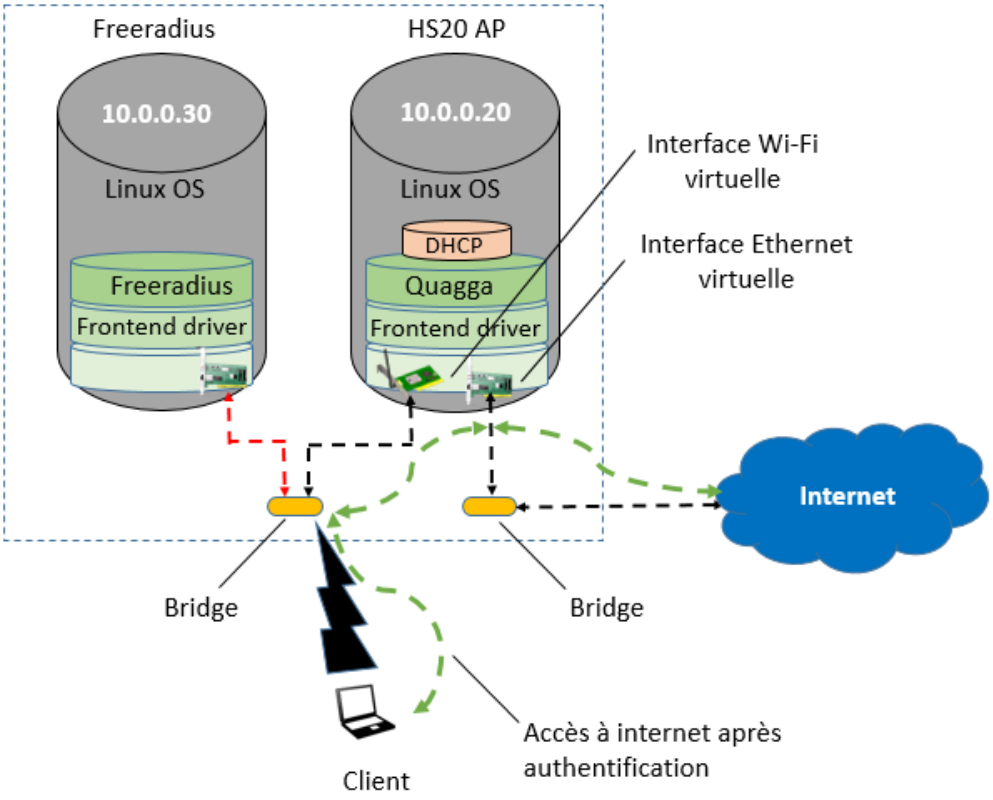

Figure 21 : Architecture Hotspot2.0 virtuel avec client authentifié

Cette solution de virtualisation de point d'accès Wi-Fi Hotspot2.0 fût présentée lors de la démonstration [9]. La démonstration consistait en une instanciation à la volée de différents points d'accès Wi-Fi implémentant différents niveau de sécurité : Open, WEP, WPA-PSK, WPA2-Enterprise. Les participants du workshop ont pu s'y connecter pour vérifier le bon fonctionnement de ceux-ci. Quant à la démonstration du point d'accès Wi-Fi Hotspot2.0, elle nécessitait qu'un de nos clients Hotspot2.0 développé s'y connecte, du fait que les équipements mobiles de configuration d'usine ne supportent pas encore une telle technologie.

2.2.4.6 Récapitulatif et avantages de la solution

A travers cette contribution, nous avons contribué à enrichir le type de services et machines virtuelles pouvant être instanciées dans la MNetBox commercialisée par l'entreprise VirtuOR. L'un de nos client cible est l'opérateur mobile, à travers ce cas d'utilisation, nous essayons de prouver l'utilité de l'installation de MNetBox supportant Hotspot2.0 chez l'un d'entre eux :

Un client X est en train d'utiliser son smartphone pour accéder à un service à travers son réseau cellulaire de l'opérateur Ox. Ce même opérateur Ox a établi un consortium de roaming dans le cadre de la norme Hotspot2.0 avec les opérateurs Oy et Oz. X se déplaçant, va se retrouver à proximité d'une MNetBox appartenant à Oz et où des points d'accès virtuels appartenant à l'opérateur Oz sont instanciés. Ces point d'accès virtuels au travers des trames GAS/ANQP annoncent qu'ils sont certifiés Passpoint et qu'entre autres ils appartiennent à l'opérateur Oz qui est en consortium de roaming avec Ox et Oy. L'équipement mobile de X (sans l'intervention de X) initie une étape d'association avec un point d'accès Wi-Fi Hotspot2.0 virtuel qu'on appellera APz. APz joue le rôle d'un NAS transparent et transmet la demande de connexion au serveur AAA de l'opérateur (AuC/HLR), celui-ci authentifiera X en vérifiant son identité (certificat X.509). Le serveur RADIUS après vérification, va en effet reconnaitre X comme étant un client légitime appartenant à Ox en accord d'itinérance avec Oz et donc lui donner accès à ses services. La session de connexion est ainsi ouverte et X libére une ressource radio du réseau cellulaire de son opérateur Ox et continue d'accéder à son service de façon transparente sur un réseau Wi-Fi appartenant à Oz. Le même scénario aurait eu lieu si un client Z appartenant à Oz se serait trouvait à proximité d'une MNetBox appartenant à Ox. De nombreux scenarios riches et variés sont envisageables pour l'utilisation des points d'accès certifiés Passpoint. Une mutualisation des MNetBox au niveau de plusieurs opérateurs n'est pas à exclure (une seule Box avec APx, APy et APz appartenant respectivement à Ox, Oy et Oz).

A ce jour, il n'existe aucune référence de recherche dans la littérature qui évoque un développement de topologie Hotspot2.0 « from scratch ». Dans [61] la proposition faite est d'améliorer la connexion au Wi-Fi eduroam grâce à 802.11u. Le point d'accès Wi-Fi Hotspot2.0 physique utilisé dans leurs expérimentations

est un Cisco Unified Wireless, qui intègre les fonctionnalités Passpoint. Le client utilisé quant à lui est un smartphone Samsung Galaxy S4 (premier équipement mobile certifié Passpoint également). Les travaux réalisés dans [61] consistent à configurer des équipements physiques déjà prêt à l'emploi. Notre solution quant à elle, a été entièrement développée par nos soins. De plus, notre démarche s'inscrit dans le NFV : le serveur d'authentification et le point d'accès Hotspot2.0 sont tous deux virtualisés et communiquent pour fournir un service d'accès. Ainsi, notre solution peut être instanciée à la volée et à la demande et s'inscrit dans la démarche d'urbanisation des points d'accès en apportant les avantages suivants :

- une souplesse de management et de délégation de la gestion des réseaux ;
- un faible coût dû aux équipements virtualisés qui engendre un déploiement à coût réduit de nouvelles solutions pour les réseaux informatiques ;
- un environnement qui peut héberger de multiples réseaux virtuels dont chacun est adapté à un service spécifique;
- une mise à jour rapide de tous les équipements virtualisés ;

2.3 L'utilisation du NFC dans un contexte de mobilité dans un environnement sans fil virtualisé

2.3.1 Les motivations

Comme nous l'avons vu dans l'implémentation de notre solution Hotspot2.0 virtualisée, la configuration du client est très complexe pour des utilisateurs non expérimentés. En effet, en fonction de la méthode d'authentification, des certificats, des clés privées et des différentes identités à configurer sont exigés (tunnels d'authentification [14] [15]). 802.1X (exigé par la norme Hotspot2.0) est une solution parfaite pour les réseaux d'entreprises car des administrateurs réseaux se chargent de fournir les moyens d'authentifications adéquats aux employés. Dans un réseau mobile public, les choses se complexifient. En effet, il faut être en mesure de fournir aux clients des certificats d'authentification. Il n'est tout simplement pas possible pour les utilisateurs non expérimentés de configurer eux-mêmes un équipement pour faire de l'EAP-TLS,

par exemple le client lambda ne dispose pas des connaissances techniques sur l'authentification ou l'accès au réseau. En outre, malgré une configuration appropriée du terminal mobile, les messages d'erreurs peuvent être incompréhensibles. En effet, les serveurs d'authentification retournent généralement des codes d'erreur, le cas échéant, qui sont difficiles à comprendre pour les utilisateurs. Par conséquent, Hotspot2.0 -à travers 802.1X- est une norme qui assure une sécurité accrue du réseau Wi-Fi, mais qui y ajoute une complexité de gestion non négligeable.

Nous allons voir dans ce qui suit que pour réduire cette complexité, nous proposons de distribuer les certificats avec des terminaux NFC reliés aux points d'accès Wi-Fi. Le client, avec son appareil mobile, doit se mettre à proximité du terminal NFC. Le client alors envoie des informations personnelles à la borne NFC qui la transmet au serveur RADIUS dans le réseau de l'opérateur pour générer des certificats X.509 clients. Le serveur RADIUS renvoie par la suite: le certificat client, le certificat root, la clé privée et le mot de passe privée via le même terminal NFC. Ainsi, le client sans avoir à configurer quoique ce soit se trouve prêt à se connecter sur le réseau Wi-Fi Hotspot2.0 proposé par son opérateur, ou un opérateur ayant signé des accords d'itinérance avec l'opérateur de ce client.

2.3.2 La technologie NFC

La technologie NFC, ou Near Field Communication (Communication dans un champ proche), est une technologie simple et intuitive qui permet d'utiliser un téléphone portable ou tout autre équipement à des fins innovantes. En 2004, des entreprises telles que Sony, Philips et Nokia ont créé le « Near Field Communication Forum » pour fixer des standards de développement et utiliser cette technologie émergente. NFC définit le procédé par lequel deux clients communiquent entre eux. Cette technologie est un dérivé de la technologie RFID (Radio Frequency Identification) à plus courte distance d'action (10cm en théorie, ~4 cm en réalité) qui utilise des vitesses faibles (106-414 kbps) sur une fréquence de 13,56MHz, ce qui permet alors à deux appareils de communiquer automatiquement quand ils sont proches l'un de l'autre. Pour ce faire, une petite carte NFC est introduite dans l'équipement mobile et permet de générer un champ

électromagnétique. NFC repose sur les éléments clés des normes de cartes sans contact existantes (ISO/IEC 14443 A&B et JIS-X 6319-4) [13].

De plus en plus, les smartphones sont équipés de cartes NFC [12]. Les applications de cette technologie incluent le paiement mobile, la billetterie, le contrôle d'accès [16] [17]. En outre, NFC peut être utilisé pour démarrer d'autres types de connexions telles que Bluetooth ou Wi-Fi du fait de sa simplicité d'utilisation. En effet, le NFC est souvent utilisé comme medium radio connexe par d'autres technologies pour échanger des paramètres de connexions. Nous pouvons prendre pour exemple la technologie WPS-NFC (Wi-Fi Protected Setup NFC) lancée par la Wi-Fi Alliance début 2007 pour simplifier la connexion d'un appareil à un réseau Wi-Fi. L'équipement alors équipé d'une interface NFC doit se rapprocher du point d'accès Wi-Fi équipé lui-même de la technologie NFC, pour récupérer la configuration du réseau nécessaire à la connexion Wi-Fi (l'AP transfert un profil de connexion comme vu précédemment dans le fichier wpa_supplicant.conf) [11]. Le WPS-NFC a été conçu pour autoriser n'importe quel client à proximité du point d'accès Wi-Fi de pouvoir récupérer le mot de passe nécessaire à une connexion Wi-Fi (WPA-PSK). La technologie WPS a été vivement critiquée [18] car elle permet à quiconque à proximité de l'AP d'accéder à un réseau qui est supposé être protégé par une clé de sécurité.

C'est dans cette démarche que nous nous positionnons pour récupérer des droits d'accès pour pouvoir s'authentifier sur le réseau Wi-Fi, à la seule condition que le client soit un client légitime (contrairement au WPS). C'est dans cet esprit que nous avons développé un nouveau concept d'architecture [23], permettant de récupérer des certificats X.509 pour pouvoir accéder à un réseau sécurisé 802.1X, en l'occurrence un réseau Wi-Fi Hotspot2.0.

2.3.3 La distribution de certificats X.509 clients pour accéder au Wi-Fi Hotspot2.0

2.3.3.1 La distribution de certificat dans le cas de l'EAP-TLS

Comme d'autres protocoles (SMTP-TLS [19], IMAP-TLS [20], HTTPS [21], etc.), EAP repose sur TLS pour assurer une authentification sécurisée. L'utilisation de certificats a ses avantages et inconvénients. Ils sont souvent considérés comme plus sûrs que les mots de passe. Toutefois, les opérations de

gestion qu'ils causent peuvent être fastidieuses (création, suppression, liste de révocation de certificats, etc.). Ainsi, une infrastructure de clés publiques (PKI) est nécessaire. La distribution des certificats aux clients est une contrainte qui ne devrait pas être négligée. Nous devons nous assurer que le certificat est correctement envoyé. Si un protocole non sécurisé, tel que SMTP, HTTP ou FTP, est utilisé pour envoyer le fichier sur Internet, la sécurité du certificat peut être compromise. De nos jours, les moyens sécurisés pour distribuer un certificat exige l'utilisation de HTTPS, SSH ou du matériel (clé USB, disque dur externe, etc.). On ne trouve aucune technique particulière dans la littérature pour distribuer un certificat électronique pour le Wi-Fi 802.1X-EAP-TLS.

Cette problématique est plus que jamais, un point critique pour le déploiement de points d'accès Wi-Fi Hotspot2.0. Les opérateurs, les fournisseurs de services Internet (FSI) et tous les acteurs des accords d'itinérances devront faire face à l'énorme tâche de distribuer les outils d'authentification à leurs abonnés utilisant l'EAP-TLS.

2.3.3.2 L'utilité du NFC dans un contexte sécurisé

Lors de l'utilisation de NFC, deux objectifs sont à atteindre : la distribution des certificats au client et l'installation de ces certificats dans l'élément sécurisé de l'appareil mobile. Ainsi l'acquisition des droits d'accès se fait en une seule phase, il n'y a plus le besoin de télécharger puis installer les certificats manuellement. La taille d'un certificat X.509 est de l'ordre de la centaine d'octets, la faible bande passante autorisée par la NFC (414Kb/s) n'est donc pas un handicap pour notre architecture. Bien au contraire, la facilité de l'opération correspond parfaitement à la philosophie de mobilité pour les smartphones. L'objectif n'est pas de rivaliser avec Bluetooth ou Wi-Fi, mais de donner une fonctionnalité simple aux appareils pour l'installation de certificats afin d'accéder à leurs services avec les outils d'authentification requis.

La très courte portée de la technologie NFC fournit déjà un aspect important de sécurité. Mais la technologie NFC n'est pas immunisée contre les attaques telles que l'écoute, la corruption, la manipulation et l'interception de données [22]. Le chiffrement n'est pas obligatoire dans la norme NFC (c'était intentionnel pour assurer que la technologie soit compatible avec les

implémentations précédentes de RFID), mais le cryptage standard AES peut être utilisé. Bien que les attaques sur NFC soient rares et nécessitent des équipements sophistiqués, elles existent. Un tel aspect de sécurité doit être traité avec la plus grande attention dans la distribution de certificats, sans laquelle toute la chaîne de sécurité (WPA2-Enterprise) établie par Hotspot2.0 peut être corrompue en amont. Il est instinctif de penser qu'un échange de données entre un appareil mobile et un terminal NFC pour initier une communication basée sur EAP-TLS avec un point d'accès Wi-Fi devrait être sécurisé par TLS. LLCPS est un draft du groupe de travail TLS de l'IETF qui décrit la mise en œuvre du protocole TLS [42] sur la couche LLCP [43] de NFC. LLCPS offre une sécurité accrue pour la communication NFC point à point [44]. LLCPS s'annonce être la solution qui permettra au NFC d'avoir un niveau de sécurité approprié pour l'échange des données personnelles et des certificats d'accès au réseau.

2.3.3.3 L'architecture proposée

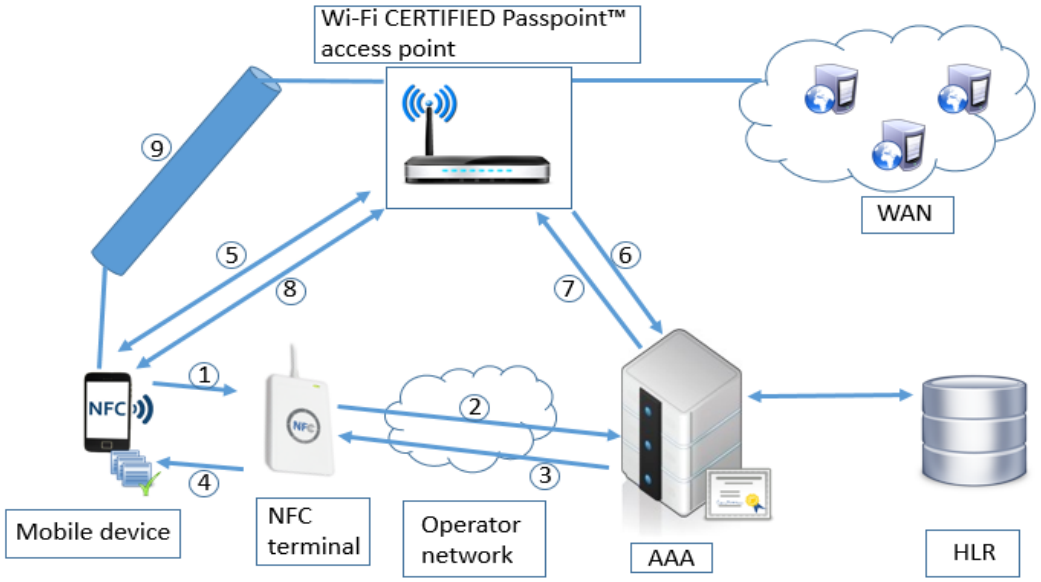

Figure 22 : Approvisionnement des certificats pour une authentification EAP-TLS à travers une borne NFC

Comme le montre la Figure 22, l'appareil mobile d'un client demande l'accès pour la première fois à un réseau Wi-Fi Hotspot2.0 fourni par son

opérateur se basant sur EAP-TLS. Le client doit obtenir un certificat client, un certificat root et une clé pour l'authentification. Le terminal NFC, appartenant à l'opérateur, peut fournir le certificat automatiquement sans aucune manipulation coté client. L'architecture proposée, propose ainsi une borne NFC pour accéder au PKI (Public Key Infrastructure : ensemble des procédures pour gérer le cycle de vie des certificats numériques). Les étapes du processus de distribution des certificats dans la Figure 22 sont les suivants:

1: L'appareil envoie son CSR (Certificate Signing Request) contenant l'IMSI du client (International Mobile Subscriber Identity) à partir de la carte SIM, ou un pseudonyme qui a déjà été accordé lors d'une authentification précédente, tel que le TMSI (Temporary Mobile Subscriber Identity), à la borne NFC.

2: La borne NFC se charge d'envoyer le CSR de l'utilisateur à l'autorité d'enregistrement (RA). Le RA est l'entité qui vérifie que les demandeurs ou les porteurs de certificat sont identifiés, que leur identité est authentique et que les contraintes liées à l'usage d'un certificat sont remplies, tout cela conformément à la politique de certification. Le RA est inclut dans le serveur AAA.

3: Le RA vérifie l'identité du client à travers son IMSI, si l'IMSI correspond à un utilisateur légitime (présent dans la base de donnée du serveur d'authentification), le RA génère un couple (clé publique, clé privée) et transmet le CSR à l'autorité de certification (CA) accompagné la clé publique. Le CA certifie la clé publique et appose sa signature sur le certificat, il délivre enfin le certificat client ainsi crée, avec le certificat root, et le couple (clé privé/ clé publique) initialement crée par le RA, à travers la borne NFC. L'ensemble de ces données peuvent être envoyé en un seul et même fichier de format PKCS#12.

4: Le terminal NFC envoie vers l'appareil mobile le fichier PKCS#12 contenant tous les outils d'authentification qui seront installés (et protégés) dans le « secure element » du mobile, ainsi qu'un profil du réseau Wi-Fi Hotspot2.0 (description wpa_supplicant.conf).

5: L'AP Wi-Fi Hotspot2.0 diffuse en continu des trames GAS/ANQP qui seront écoutées par le client lors de la découverte et pré-association 802.11u du réseau. Si les capacités de l'AP correspondent aux besoins du client, une demande d'association est envoyée et l'authentification se basant sur EAP-TLS peut commencer.

6: L'AP envoie une demande d'authentification du client avec son certificat (EAP-TLS) au serveur AAA pour vérifier si l'utilisateur est légitime. Une session EAP est ainsi créée entre le client et le serveur pour l'authentification (Cf. Figure 14) via les certificats délivrés par la borne NFC.

7: La négociation TLS est terminée avec succès (EAP SUCCESS), à la fois le serveur et le client ont été authentifiés dans l'échange.

8: Des clés uniques sont distribuées (PMK : Pairwise Master Key) par le serveur d'authentification, au point d'accès et à l'appareil pour chiffrer le trafic Wi-Fi.

9: L'appareil mobile a accès à Internet sur un point d'accès Wi-Fi Hotspot2.0 publique avec une sécurité WPA2-Enterprise.

Il est important de mentionner que dans l'étape 8, la distribution des clés de chiffrement au client et au point d'accès se fait après que l'authentification soit réussie. Le serveur d'authentification se charge de délivrer une Master Key (MK) au client mobile, et une Pairwise Master Key (PMK) au point d'accès. Le client mobile dérive sa clé pour obtenir la PMK à son tour. Le client et l'AP se lancent ensuite dans un 4-way handshake pour dériver le PMK en PTK (Pairwise Transient Key) pour l'unicast et GTK (Group Transient Key) pour le broadcast/multicast.

2.3.4 La création d'un point d'accès Wi-Fi virtuel sécurisé au travers d'une borne NFC

Comme nous l'avons vu, l'utilisation d'une borne NFC couplée à un point d'accès Wi-Fi Hotspot2.0 permettait d'envisager une méthode de distribution de certificats électroniques permettant à un client de s'authentifier au réseau Wi-Fi. Dans la même philosophie, une autre méthode à base de bornes NFC a été proposée pour permettre à un client de créer son propre point d'accès Wi-Fi virtuel (non Hotspot2.0) sécurisé.

2.3.4.1 L'automatisation

Dans le Chapitre 1, nous avons énuméré les nombreux avantages des points d'accès Wi-Fi virtuels. L'idée principale de notre approche est de trouver un mécanisme qui induit la création d'un point d'accès Wi-Fi virtuel et qui dans le même temps donne au client toutes les politiques de sécurité nécessaires afin de se connecter à son AP sans intervention humaine. La problématique qui se pose est de savoir comment un utilisateur peut-il créer son propre point d'accès Wi-Fi virtuel sécurisé dans un lieu public présentant l'infrastructure de virtualisation adéquate ?

Notre étude se penche sur le cas d'un AP virtuel implémentant WPA2-Entreprise (authentification 802.1X et chiffrement AES). Pour la méthode d'authentification 802.1X, nous choisissons comme précédemment l'EAP-TLS qui, nous le rappelons, est considéré comme la méthode EAP la plus robuste mais aussi la plus complexe à mettre en oeuvre. Dans un réseau sécurisé, les parties communicantes doivent évidemment se faire mutuellement confiance. Une méthode pour l'établissement de cette confiance passe par un tiers de confiance, par exemple, une autorité de certification (CA) telle une infrastructure de clés publiques ou PKI (Public Key Infrastructure). Dans la pratique, avec EAP-TLS, nous devons configurer les certificats pour le serveur et le client, afin d'assurer l'authentification mutuelle. Ces certificats doivent être signés par une autorité de certification (serveur AAA). La solution proposée consiste à automatiser en un seul geste (se rapprocher de la borne NFC) à la fois la création d'un AP Wi-Fi WPA2-Enterprise virtuel, et la distribution des certificats adéquats au client pour

se connecter à celui-ci. Cette méthode permet au client d'interagir avec un hardware permettant la virtualisation et de crée son propre accès sans avoir à configurer son équipement. Cette contribution a fait l'objet d'un papier journal [24].

2.3.4.2 La distribution du certificat et la création du point d'accès

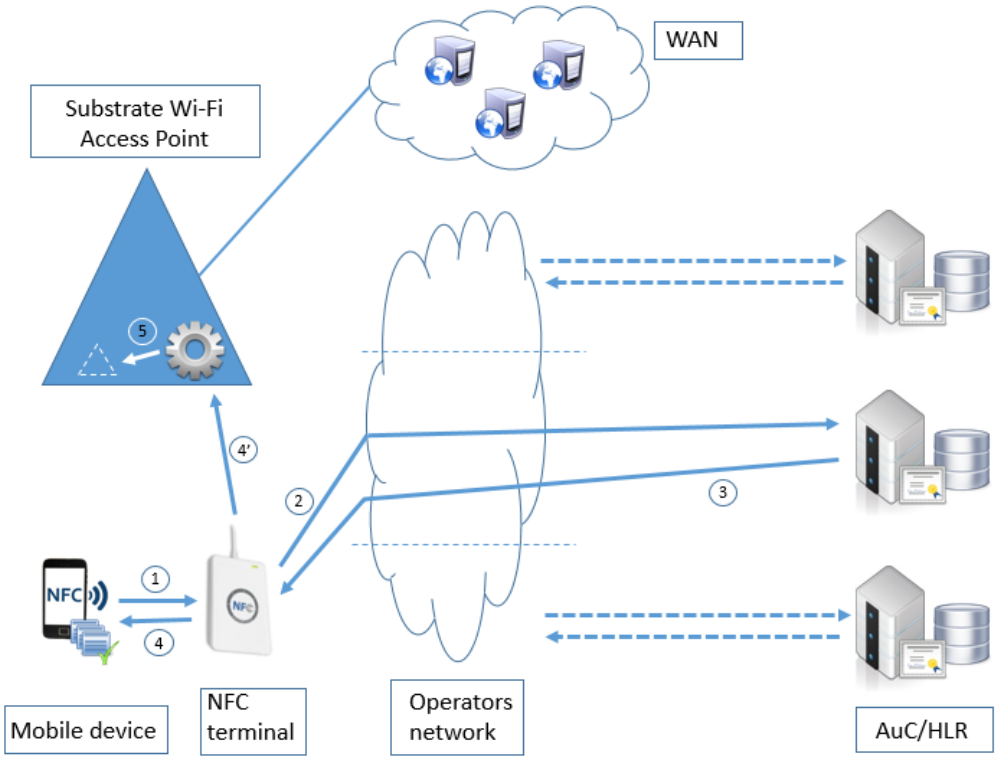

Figure 23 : Distribution de certificat et création d'un AP Wi-Fi virtuel via NFC

Comme le montre la Figure 23, un équipement client interagit avec une borne NFC rattachée au point d'accès Wi-Fi physique fourni par son opérateur, pour créer un point d'accès Wi-Fi virtuel sécurisé 802.1X-EAP-TLS et pour obtenir un certificat client pour l'authentification. Les étapes sont les suivantes:

1: L'appareil envoie son CSR (Certificate Signing Request) contenant l'IMSI du client (International Mobile Subscriber Identity) à partir de la carte SIM, ou un pseudonyme qui a déjà été accordé lors d'une authentification précédente, tel que le TMSI (Temporary Mobile Subscriber Identity), à la borne NFC.

2: La borne NFC se charge d'envoyer le CSR de l'utilisateur à l'autorité d'enregistrement (RA). Le RA est l'entité qui vérifie que les demandeurs ou les porteurs de certificat sont identifiés, que leur identité est authentique et que les contraintes liées à l'usage d'un certificat sont remplies, tout cela conformément à la politique de certification. Le RA est inclut dans le serveur AAA.

3: Le RA vérifie l'identité du client à travers son IMSI, si l'IMSI correspond à un utilisateur légitime (présent dans la base de donnée du serveur d'authentification), le RA génère un couple (clé publique, clé privée) et transmet le CSR à l'autorité de certification (CA) accompagné la clé publique. Le CA certifie la clé publique et appose sa signature sur le certificat, il délivre enfin le certificat client ainsi crée, avec le certificat root, et le couple (clé privé/ clé publique) initialement crée par le RA, à travers la borne NFC. L'ensemble de ces données peuvent être envoyé en un seul et même fichier de format PKCS#12. Le serveur d'authentification autorise le nouveau point d'accès Wi-Fi virtuel à communiquer avec lui (fichier clients.conf dans Freeradius) pour permettre l'authentification avenir du client.

4: Le terminal NFC envoie vers l'appareil mobile le fichier PKCS#12 contenant tous les outils d'authentification qui seront installés (et protégés) dans le « secure element » du mobile, ainsi qu'un profil du réseau Wi-Fi virtuel crée (description wpa_supplicant.conf).

4 ': La borne NFC envoie des informations au point d'accès physique sur le point d'accès Wi-Fi virtuel en accord avec les informations d'authentification du client (méthode EAP-TLS) pour créer un fichier de configuration hostapd.conf en accord avec le profile crée dans wpa_supplicant.conf au niveau du client.

5: Le point d'accès Wi-Fi virtuel est créé à l'aide d'un script shell permettant de créer une machine virtuelle sur l'hyperviseur et de l'activation de l'interface Wi-Fi virtuelle correspondante. Le point d'accès Wi-Fi virtuel WPA2-Enterprise ainsi crée se met en marche et commence à envoyer des beacons pour annoncer sa présence.

2.3.4.3 L'authentification et la connexion du client

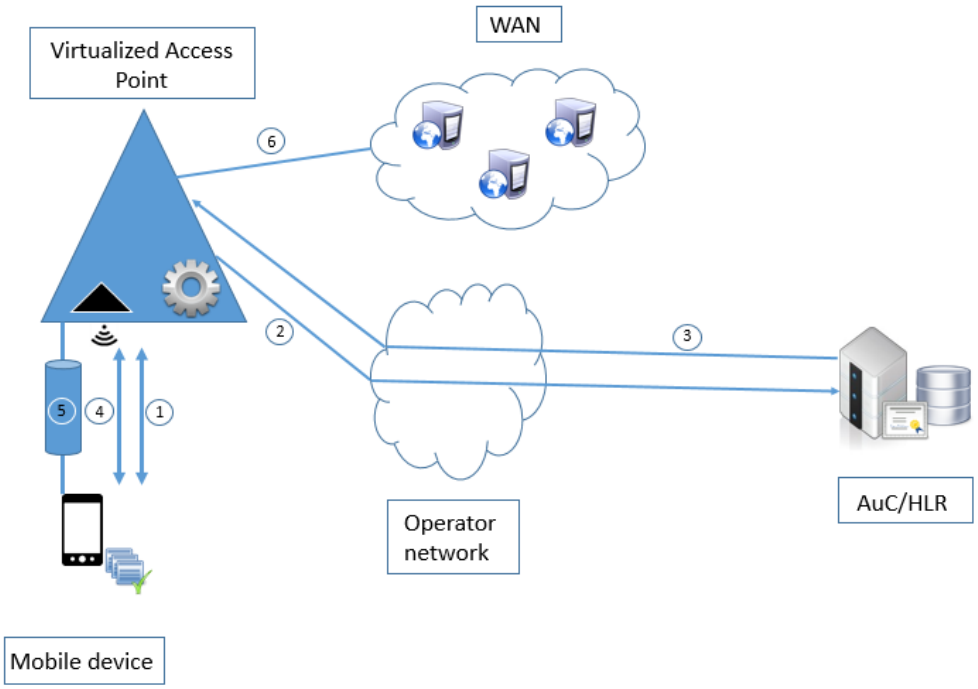

Figure 24 : Client se connectant à un point d'accès Wi-Fi sécurisé WPA2-Enterprise créé via NFC

Après création du point d'accès Wi-Fi virtuel, le client dispose de certificats et d'une politique d'accès au réseau Wi-Fi. Il est en mesure d'accéder à ses services par le biais de l'AP virtuel sécurisé. Comme l'illustre les étapes de la Figure 24 :

1: Le client s'associe à l'AP. Toutefois, il n'est pas encore autorisé à envoyer des données. Il doit tout d'abord s'authentifier, pour se faire il initie une session EAP-TLS.

2: L'AP envoie une demande d'authentification du client avec son certificat (EAP-TLS) au serveur AAA pour vérifier si l'utilisateur est légitime. Une session EAP est ainsi créée entre le client et le serveur pour l'authentification via les certificats délivrés par la borne NFC.

3: La négociation TLS est terminée avec succès (EAP SUCCESS), à la fois le serveur et le client ont été authentifiés dans l'échange EAP. Un PMK est partagé entre le client et l'AP.

4: De nouvelles clés de cryptage sont dynamiquement dérivées du PMK pendant un 4-way handshake: PTK pour l'unicast et GTK pour le multicast/broadcast.

5: Les échanges de données sont désormais cryptés entre l'appareil mobile et l'AP.

6: Le client a désormais accès à internet avec un point d'accès Wi-Fi virtuel sécurisé qu'il vient de créer sur le substrat physique de son opérateur.

2.4 Conclusion

Nous venons de voir que Hotspot2.0 permet d'offrir une solution globale pour un accès transparent sur les réseaux Wi-Fi. En outre Hotspot2.0 permet de faciliter le délestage du réseau 3G/4G de l'opérateur et permet aussi d'assurer l'itinérance des abonnés entre opérateurs. Cette nouvelle technologie permettra d'améliorer de façon significative l'expérience utilisateur en automatisant la connexion sécurisée aux hotspots. Le challenge relevé par Hotspot2.0 est de rendre les réseaux Wi-Fi publiques aussi sécurisés que les réseaux cellulaires. Plus que jamais, l'accès sans fil est entrain de métamorphoser le design des architectures réseaux existantes. Avec une mobilité accrue des clients, l'accès au réseau en lui-même constitue un service; le point d'accès Wi-Fi doit suivre le client partout. Les progrès des technologies de virtualisation ont créé une nouvelle opportunité pour les opérateurs de profiter des ressources du réseau. Dans notre approche, nous essayons d'adapter les concepts de virtualisation pour répondre à un grand nombre de défis en matière de télécommunications et d'urbanisation de réseaux. La virtualisation des points d'accès Wi-Fi Hotspot2.0 que nous avons réalisé pendant nos travaux ouvre de nouveaux horizons sur la mobilité du client mais aussi des points d'accès (puisqu'ils sont logiciels).

Nous avons aussi vu que l'un des problèmes posé par Hotspot2.0 est la nécessité pour le client d'avoir les outils adéquats pour s'authentifier. Nous avons vu par exemple que dans le cas d'une authentification 802.1X-EAP-TLS, le client devra rapatrier ses certificats d'un serveur d'authentification, pour ensuite les installer sur son équipement mobile. Cette tâche est fastidieuse et complexe. Il est quasiment impossible pour un utilisateur non expérimenté de configurer lui-même un équipement pour faire de l'EAP-TLS. Nous avons alors proposé une architecture à base de bornes NFC pour faciliter la distribution des certificats pour permettre aux clients légitimes d'accéder à un réseau Wi-Fi Hotspot2.0 d'un simple geste. Nous avons aussi vu qu'il était possible pour un client d'interagir avec un substrat physique de virtualisation (tel que la MNetBox) pour créer un réseau Wi-Fi à la demande via une borne NFC rattachée au substrat. Cette dernière approche vient apporter une réponse à la problématique évoquée en fin de section 1.6 dans le Chapitre 1.

Chapitre 3 : SDN et points d’accès Wi-Fi virtuels

3.1 Introduction

L'explosion de la vente des appareils mobiles, de la taille des contenus numériques, de la virtualisation des serveurs, et l'avènement des services de cloud computing sont parmi les tendances conduisant l'industrie des réseaux à réexaminer les architectures de réseaux traditionnels. La plupart des réseaux classiques sont hiérarchiques, construits avec des rangées de switchs Ethernet disposés dans une topologie en arbre (tree topology). Cette conception des réseaux avait un sens quand le modèle client-serveur était dominant. Toutefois, une telle architecture statique est mal adaptée aux besoins de calcul et de stockage dynamique des data centers d’entreprise, de campus, et des réseaux opérateurs. Parmi les tendances informatiques clés qui déterminent la nécessité d'un nouveau paradigme de réseau, nous pouvons citer:

- **Le changement des schémas du trafic**: Dans les data centers d'entreprise, les modèles de trafic ont changé de manière significative. Contrairement aux applications client-serveur où la majeure partie de la communication se produit entre un client et un serveur, les applications d'aujourd'hui accèdent à différentes bases de données et serveurs, créant une rafale de trafic "est-ouest" de machine à machine avant de retourner les données à l’utilisateur final dans le modèle de trafic classique "nord-sud". Les utilisateurs eux-mêmes sont en train de changer les modèles du trafic réseau en demandant l'accès au contenu et aux applications de leurs entreprises, à travers toute sorte d’équipements (consumérisation de l’IT), n'importe où, et à tout moment.
- **L’émergence des services cloud** : les entreprises ont adopté avec enthousiasme le cloud computing public et privé, résultant en une croissance sans précédent de ces services. Les utilisateurs finaux nécessitent maintenant la souplesse d’accès à ces applications et

l'infrastructure sous-jacente nécessaire. La complexité est d'autant plus accrue car ces infrastructures nécessitent des nouvelles exigences en termes de sécurité, de conformité et de disponibilité. Aussi comme nous l'avons déjà vu dans le chapitre précédent, la distribution des droits d'accès en libre-service, que ce soit dans un cloud privé ou public, exige l'élasticité de calcul, de stockage et des ressources réseau, idéalement avec des outils standards.

- **L'émergence du Big Data** : La manipulation des Big Data d'aujourd'hui ou des « méga bases de données » nécessite un traitement massif et parallèle sur des milliers de serveurs, qui doivent tous être connectés entre eux. La hausse du volume des données alimente une demande constante d'augmentation des capacités du réseau.

En parlant de ces nouveaux challenges et tendances informatiques, instinctivement seul le cœur du réseau semble être touché par ce phénomène. En effet c'est au niveau des data centers (contenant les VMs) que des innovations ont vu le jour pour modifier le comportement des réseaux traditionnels. Or dans nos travaux, nous avons vu que la virtualisation des réseaux peut être ramenée en périphérie du réseau (sur l'accès), notamment avec les points d'accès Wi-Fi virtuels que nous avons développés. Nous savons aussi que les réseaux Wi-Fi sont laborieux à gérer, car ils sont soumis à des erreurs de configuration et des retards importants dans le dépannage et l'approvisionnement, en raison d'un processus manuel intensif. En outre, l'introduction de nouveaux services prend toujours un certain temps (plusieurs semaines) en raison de l'activation manuelle du service, l'assurance et la livraison: les architectures Wi-Fi ne sont pas flexible [33]. De plus, les architectures Wi-Fi traditionnelles basées sur les contrôleurs sont généralement fermées et non programmables. En d'autres termes, la plupart des fournisseurs utilisent des protocoles propriétaires pour la communication entre le contrôleur et les points d'accès. Ce manque d'interopérabilité entre les produits de différents fournisseurs peut être problématique [34].

Dans ce contexte, le marché du LAN sans fil reste extrêmement compétitif. Des fournisseurs tels que Cisco, Aruba (récemment acquis par HP), Ruckus et Meru se bousculent pour prendre des parts de marché. Dans cet environnement qui favorise l'innovation, SDN (Software-Defined Networking)

semble être l'un des paradigmes pour optimiser le déploiement et la gestion des réseaux Wi-Fi.

Dans ce chapitre nous allons présenter la solution apportée par le SDN, qui a été créé pour ajouter un niveau d'abstraction de gestion globale aux fonctionnalités des équipements réseau. Ensuite, nous allons voir l'intérêt d'introduire SDN sur le réseau d'accès, et comment sommes-nous parvenu à l'intégrer dans notre environnement Wi-Fi virtualisé.

3.2 SDN

Le Software Defined Networking (SDN) est une architecture de réseau émergente où le plan de contrôle du réseau est découplé du plan de données [25]. Ce découplage autorise le déploiement du plan de contrôle sur des plateformes de plus grandes capacités que celles des commutateurs réseaux traditionnels. La Figure 25 montre une vue logique de l'architecture SDN. L'intelligence du réseau est (logiquement) centralisée dans les logicielles des contrôleurs SDN, qui maintiennent une vision globale du réseau. Par conséquent, le réseau apparaît pour les applications comme un commutateur logique et unique. Avec SDN, les opérateurs et les entreprises peuvent prendre le contrôle du réseau avec une seule entité (contrôleur SDN) indépendamment des vendeurs d'équipements réseaux qui ont tendance à fermer leurs infrastructures par des solutions propriétaires. L'une des innovations majeures de SDN est la simplification des équipements de réseau eux-mêmes. Ces derniers n'ont plus besoin de comprendre et de traiter une multitude de protocoles, mais simplement accepter les instructions fournis par les contrôleurs SDN.

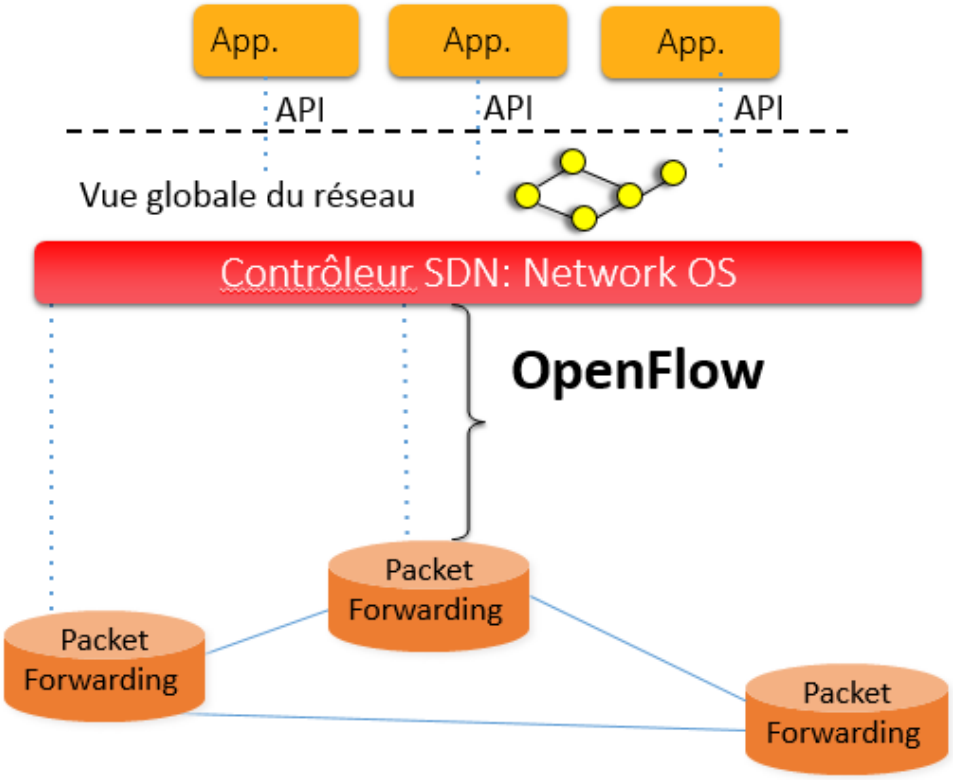

Figure 25 : Architecture SDN

SDN permet aux opérateurs de réseaux et aux administrateurs de configurer par programmation cette abstraction de réseau simplifié [26], plutôt que d'avoir à manipuler des dizaines de milliers de lignes de configuration dispersés parmi des milliers de dispositifs. En outre, cette nouvelle architecture permet de tirer parti de l'intelligence centralisée du contrôleur SDN, en permettant la modification du comportement du réseau en temps réel et de déployer de nouvelles applications et services de réseau en un temps record. En centralisant l'état du réseau dans la couche de contrôle, SDN donne aux gestionnaires de réseau la flexibilité pour configurer, gérer, sécuriser et optimiser les ressources de réseau via des programmes SDN dynamiques et automatisés. De plus, les gestionnaires de réseaux peuvent écrire eux-mêmes ces programmes SDN, au lieu d'attendre l'intégration de nouvelles caractéristiques par les vendeurs d'équipements dans des environnements logiciels propriétaires et fermés.

En plus de l'abstraction du réseau, les architectures SDN supportent un ensemble d'API qui permettent de mettre en œuvre des services de réseau, y compris le routage, le multicast, la sécurité, le contrôle d'accès, la gestion de la bande passante, l'ingénierie de trafic, la qualité de service, et toutes les formes de

gestion de politiques, taillées sur mesure pour répondre aux objectifs des entreprises. Par exemple, SDN facilite la définition et l'application des politiques d'accès cohérentes à travers les connexions filaires et sans fil sur un campus [28], et c'est sur ce dernier point que nous avons focalisé notre axe de recherche.

3.3 L'utilisation des SDN dans les réseaux sans fils

De nos jours, peu d'équipements de réseau sur le marché supportent SDN. Néanmoins, cela évolue rapidement surtout en ce qui concerne les points d'accès Wi-Fi où certains fournisseurs l'intègrent [28] [29]. En observant la croissance du nombre de points d'accès Wi-Fi et l'émergence de SDN, il est légitime de supposer que dans quelques années, la plupart des AP Wi-Fi seront SDN.

Le monde des réseaux Wi-Fi a toujours été problématique, à cause des solutions propriétaires imposées par les vendeurs, rendant difficile toute interopérabilité. Le pilotage des points d'accès Wi-Fi n'est pas quelque chose de nouveau. En effet, les vendeurs utilisent souvent leurs propres protocoles de configuration à distance à partir d'un WLC (Wireless LAN Controller). Les WLC commercialisés par les grandes industries du Wi-Fi (Cisco, Juniper, Ruckus, etc.) font partis d'offres commerciales d'équipements uniquement compatibles avec le reste des produits de la gamme (les AP Wi-Fi). Avec cette technique, les fournisseurs forcent les clients à utiliser les équipements d'une marque unique pour l'ensemble de leur architecture. D'autres protocoles non propriétaires ont émergé tels que CAPWAP (Control And Provisioning of Wireless Access Points) [30] pour standardiser la communication entre les contrôleurs Wi-Fi et les AP. Ce protocole développé par l'IETF (les auteurs du protocole travaillant entre autre pour Cisco et Aruba) a rapidement connu des implémentations propriétaires, rendant de nouveau les équipements non interopérables à cause des extensions fournisseur.

L'émergence de SDN et son application aux points d'accès Wi-Fi ouvre de nouvelles opportunités pour construire des architectures hétérogènes contrôlées à partir d'un seul contrôleur. La section suivante présente OpenFlow qui est l'un des protocoles qui définit la communication entre le plan de contrôle et de données. Nous verrons ainsi que OpenFlow est l'une des solutions prometteuses au

remplacement de CAPWAP. Par la suite nous présenterons notre implémentation d'une telle solution dans un environnement de points d'accès Wi-Fi virtuels.

3.4 Point d'accès Wi-Fi virtuel OpenFlow

3.4.1 OpenFlow

Bien qu'il ait été conçu à l'origine pour l'expérimentation de réseau, OpenFlow est maintenant mis en œuvre par divers fournisseurs de matériel tels que Juniper, IBM et HP [31]. OpenFlow ne repose sur aucune fonctionnalité fermée ou propriétaire. Il fait abstraction de la couche matérielle sous-jacente et vise à être adopté par un grand nombre de fournisseurs d'équipements. Le protocole permet à un point d'accès Wi-Fi (ou commutateur originalement) OpenFlow d'être configuré par un contrôleur SDN supportant OpenFlow. Un point d'accès Wi-Fi et un contrôleur peuvent s'envoyer des messages OpenFlow à travers un canal SSL (tunnel TLS) ce qui permet un chiffrement de bout en bout des données, mais aussi une authentification mutuelle entre l'AP et le contrôleur en utilisant des certificats X.509 (Figure 26).

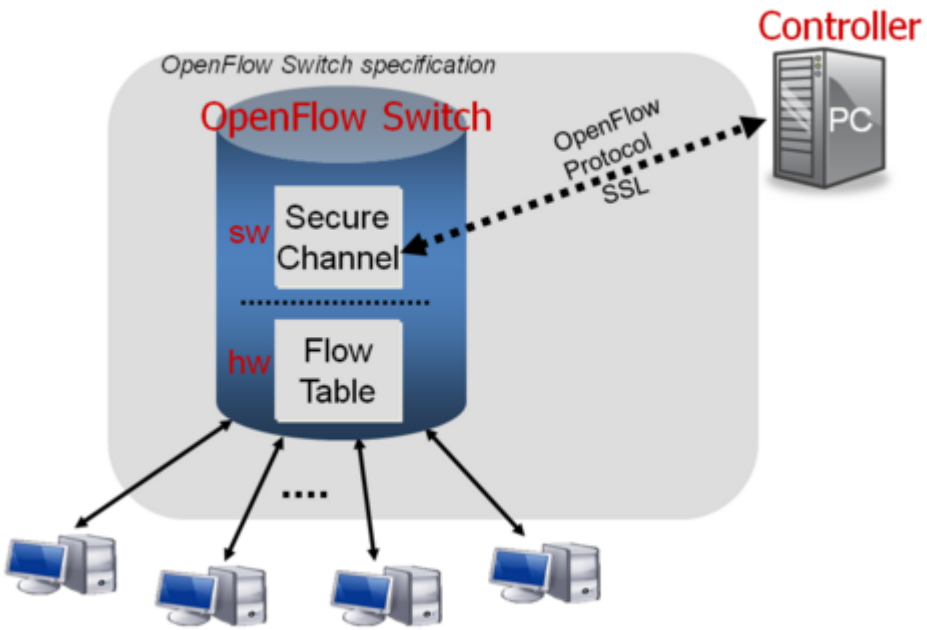

Figure 26 : Communication entre un commutateur OpenFlow et son contrôleur SDN

Grâce à sa vision complète du réseau, le contrôleur OpenFlow envoi les règles d'acheminements des paquets aux points d'accès Wi-Fi, qui les enregistrent alors dans leurs tables de routage respectives. Les points d'accès Wi-Fi OpenFlow se basent sur le concept de table de flux (à un flux correspond une règle), dans laquelle toutes les règles envoyées par le contrôleur sont enregistrées. Si le point d'accès Wi-Fi doit envoyer un flux auquel aucune correspondance n'existe dans la table de flux, il se redirige vers le contrôleur pour lui envoyer une « table miss action » en envoyant la totalité ou une partie du paquet concerné. Les flux en question sont très hétérogènes et peuvent aussi bien concerner Ethernet, VLAN 802.1Q, MPLS, IP, TCP et UDP [32].

A titre d'exemple, les règles dans les tables de flux sont:

- Jeter les paquets du flux. (DROP);
- Envoyer les paquets du flux sur un port particulier de l'équipement (SEND);
- Limiter la bande passante;
- Modifier l'entête, (par exemple IP, VLAN).

3.4.2 OpenFlow et CAPWAP

Comme nous l'avons mentionné précédemment, OpenFlow apporte un moyen de configurer le chemin des données dans un élément de réseau (point d'accès Wi-Fi, commutateur, routeur, etc.) à partir d'un contrôleur externe. Dans le monde du Wi-Fi, le protocole CAPWAP est censé faire la même chose sauf qu'il est spécifique aux points d'accès Wi-Fi.

Une mise en parallèle du fonctionnement des deux protocoles montre que le mode opératoire est le même. Le contrôleur Wi-Fi OpenFlow (ou contrôleur CAPWAP) envoie des messages aux points d'accès Wi-Fi sur le chemin que les flux de données doivent prendre, quelle action utiliser, quand changer l'action. OpenFlow ne définit pas encore des règles spécifiques au Wi-Fi et n'a pas encore l'équivalent des champs et des attributs dans les messages CAPWAP (puissance radio, roaming d'un AP à un autre, etc.). Le protocole CAPWAP quant à lui est orienté Wi-Fi. Par exemple, dans les architectures classiques, une fois que le point d'accès détecte que la force du signal du client est trop faible, il va envoyer un message au contrôleur CAPWAP pour que l'utilisateur mobile quitte la zone de

couverture. À ce stade, le contrôleur CAPWAP enverra au client mobile (par le même AP) des recommandations sur le nouvel AP à sélectionner. Une fois que l'utilisateur mobile se connecte à un autre point d'accès avec un meilleur signal, l'AP va envoyer un message via le protocole DTLS (Datagram Transport Layer Security) pour que le contrôleur CAPWAP vérifie les informations d'identification du client. Après ces étapes, le client se connecte au nouveau point d'accès Wi-Fi. Bien que le client soit connecté à un nouveau point d'accès, le réseau Wi-Fi n'est pas encore prêt pour l'itinérance: les commutateurs Wi-Fi ont encore de vieilles règles de routage et la reconfiguration des tables de routage introduit des retards supplémentaires pour le client [30].

De nombreuses propositions ont été faite sur comment introduire SDN sur les réseaux Wi-Fi pour éviter de dissocier les règles de mobilité d'un AP à un autre, et l'acheminement des données durant cette mobilité comme l'explique la Figure 27. AeroFlux [35] est une architecture basée sur le Wi-Fi SDN qui recueille les caractéristiques des liens sans fil sur deux niveaux de contrôle (Global Controller et Near-Sighted Controller) SDN. Cette approche est intéressante mais ne décrit pas l'ensemble complexe des réseaux Wi-Fi impliquant l'itinérance dans un environnement multi-AP. Chandelle [36] est un système proposant une architecture dans un contexte multi-AP pour faire face aux limitations de CAPWAP à travers les réseaux SDN pour une itinérance rapide et souple. La solution proposée utilise des AP Wi-Fi traditionnels reliés à des commutateurs OpenFlow physiques qui introduisent un délai supplémentaire. La démarche est très intéressante, mais développer un point d'accès Wi-Fi OpenFlow en un seul bloc (le point d'accès Wi-Fi supporte OpenFlow) serait plus approprié.

La problématique reste ouverte et aucune solution standardisée n'a encore vu le jour. OpenFlow et CAPWAP ne sont pas des protocoles concurrents mais complémentaires. Dans un souci de standardisation, il semblerait plus simple aux développeurs d'OpenFlow d'ajouter d'autres actions dans les messages OpenFlow pour couvrir ce que le protocole de CAPWAP essaie de faire. Notre contribution va alors résider dans le développement d'un point d'accès Wi-Fi OpenFlow, qui sera potentiellement capable d'opérer une itinérance client plus fluide, si les fonctionnalités de CAPWAP intègrent le protocole OpenFlow.

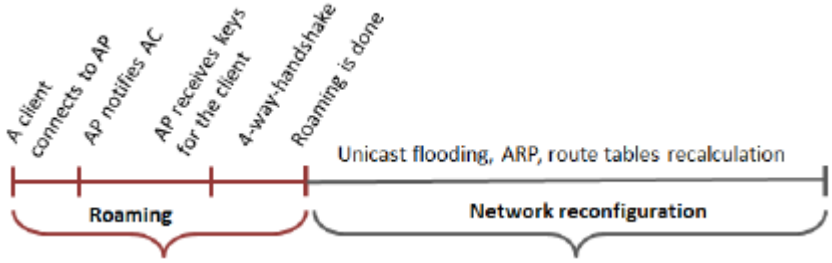

Figure 27 : Processus de reconnexion à un AP lors de l'itinérance

3.4.3 Open vSwitch

Open vSwitch [37], parfois abrégé OVS, est une implémentation open-source d'un commutateur multicouche virtuel distribué. Le but principal de l'Open vSwitch est de fournir une pile de commutation pour les environnements matériels de virtualisation, tout en supportant de multiples protocoles et normes utilisés dans les réseaux informatiques. Ce logiciel est devenu massivement utilisé par les chercheurs en SDN, car il supporte les fonctionnalités d'OpenFlow. OVS est un « couteau suisse » de la commutation, il inclut entre autres comme fonctionnalités : VLAN, le trunking 802.1q, STP 802.1D, LACP, SPAN/RSPAN, les tunnels GRE VXLAN IPSEC, etc.

Nous avons ainsi basé nos expérimentations sur ce commutateur OpenFlow logiciel que nous installons dans les machines virtuelles servant à créer nos points d'accès Wi-Fi virtuels. Nous installons et configurons la version 2.0.0 d'Open vSwitch sur une machine virtuelle qui est assez instinctive et ressemble à l'utilisation des bridges Linux natifs. Ainsi pour créer un point d'accès Wi-Fi OpenFlow virtuel, il nous suffit de créer un commutateur OpenFlow dans une machine virtuelle qui sera bridgée sur une interface Wi-Fi virtuelle (comme dans toutes nos expérimentations précédentes). La machine virtuelle quant à elle à deux interfaces : eth0 qui est reliée logiquement à l'interface Wi-Fi virtuelle sur le dom0, et eth1 qui est reliée à notre contrôleur SDN. Comme le montre la Figure 28, les deux interfaces eth0 et eth1 sont bridgées à l'intérieur de la machine virtuelle sur le bridge OVS qui implémente OpenFlow.

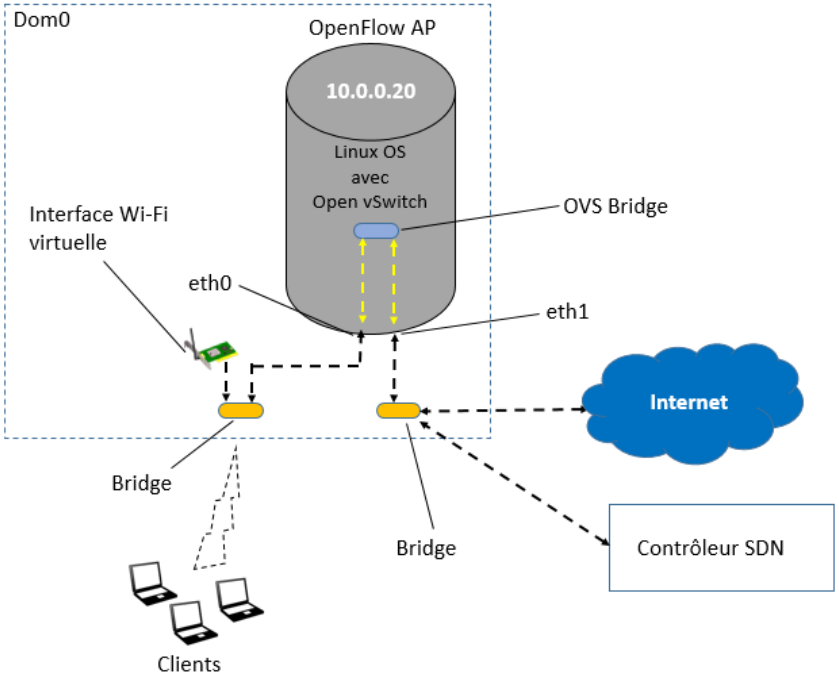

Figure 28 : Topologie réseau de notre point d'accès Wi-Fi OpenFlow virtuel

Open vSwitch se manipule au travers des commandes *ovs-vsctl,* nous avons procédé ainsi pour pouvoir construire cette topologie:

`ovs-vsctl add-br brX` ← La Création d'un bridge OVS nommé brX

Comme le montre la Figure 29, cette commande a créé un port interne rattaché logiquement à la pile protocolaire IP. Notre but est de faire passer le trafic de eth0 et eth1 à travers le bridge.

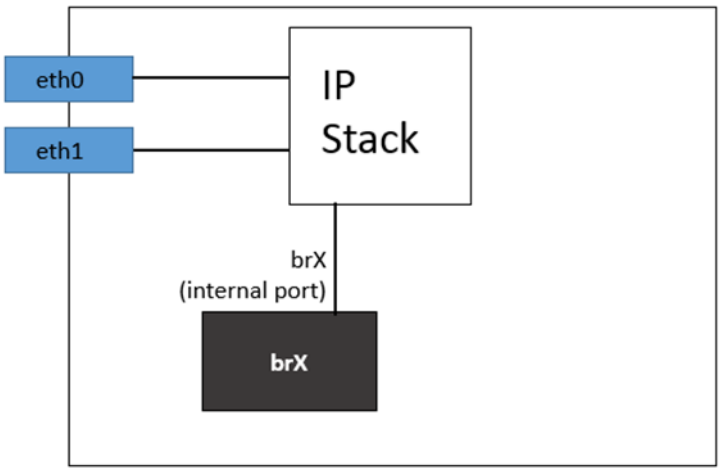

Figure 29 : Création d'un bridge OVS, interfaces non bridgées

`ovs-vsctl add-port brX eth0` ← L'ajout de l'interface sur brX

`ovs-vsctl add-port brX eth1` ← L'ajout de l'interface sur brX

```
ifconfig eth0 0
ifconfig eth1 0
dhclient brX
```

Après avoir bridgé les interfaces, il est nécessaire de leur enlever leurs adresses IP pour que brX soit l'interface qui achemine le trafic comme l'illustre la Figure 30.

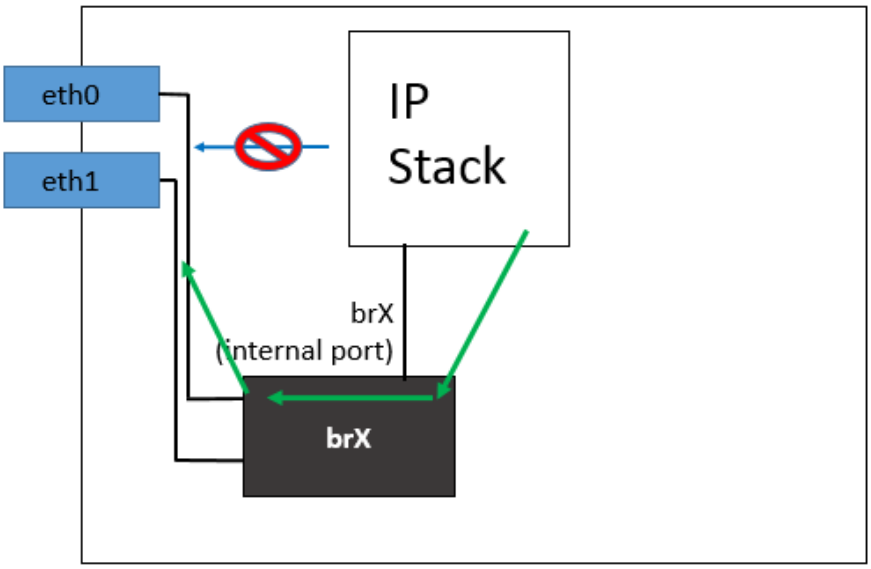

Figure 30 : Interfaces bridgées au bridge OVS

Pour voir l'état des interfaces bridgées sur le brX, on tape la commande *ovs-vsctl show* qui nous retourne comme résultat :

```
1ab8ae7e-e9da-4af9-92f6-03b5245d2544
    Bridge brX
        Port "eth0"
            Interface "eth0"
        Port "eth1"
            Interface "eth1"
        Port brX
```

```
Interface brX
    type: internal
```

Cela nous montre que les interfaces ont été correctement ajoutées sous la pile protocolaire du bridge brX. OpenFlow est activé par défaut, il ne manque plus que la création et la configuration d'un contrôleur SDN pour pouvoir établir une liaison avec le commutateur OpenFlow ainsi créé.

3.4.4 Contrôleur ONOS

Il existe une multitude de contrôleurs SDN qui ont vu le jour. Nous pouvons citer à titre d'exemple les plus connus d'entre eux: OpenDaylight, NOX, ONOS et PRISM. Dans le lot, nous avons choisi le contrôleur ONOS (Open Network Operating System) qui est le contrôleur développé par l'ONG ON.Lab (The Open Networking Lab) pour donner une alternative de substitution à OpenDaylight, ce dernier est poussé principalement par Cisco et IBM et devenant ainsi petit à petit une solution propriétaire [38]. OpenDaylight est accusé par les membres d'ON.Lab (en majorité, des pionniers de SDN) de servir les intérêts des industriels se trouvant derrière, pour configurer les équipements traditionnels déjà existant au lieu d'apporter la « plus-value SDN » [39]. ONOS a émergé alors comme le contrôleur SDN open-source soutenu par une grande communauté de chercheurs pour le développement des réseaux SDN loin des spéculations industrielles. ONOS apporte le concept de « white box » dont les autres contrôleurs SDN se sont éloignés. Ce contrôleur est l'un des plus complets concernant le clustering de contrôleurs [62]. Certes cette problématique n'a pas été étudiée dans nos travaux, mais elle constitue un enjeu d'avenir du SDN et l'interconnexion de réseaux SDN.

Pour compléter notre architecture SDN, nous décidons d'installer ONOS version 1.2.0 sur une machine virtuelle tournant sur un Linux OS. Cette machine virtuelle est instanciée sur un autre équipement physique plus puissant [40] (un RackMount VirtuOR qui est un serveur doté d'une puissance de calcul adaptée) que celui du point d'accès Wi-Fi OpenFlow virtuel. Les deux machines sont reliées par un câble FastEthernet pour établir la connexion SSL. Pour être compilé ONOS nécessite ces différents éléments :

- Java 8 JDK

- Apache Maven 3.3.1
- git
- bash
- Apache Karaf 3.0.3

Une fois installé sur la machine virtuelle, ONOS doit être configuré en installant les applications dont nous avons besoin telles que le forwarding pour activer la transmission des paquets d'un port à l'autre sur l'OVS bridge (on autorise le trafic transitant de eth0 à eth1) :

```
onos> app activate org.onosproject.fwd
```

A ce stade, nous donnons une adresse IP à notre machine ONOS virtuelle, par exemple 10.0.0.80, qui sera l'adresse IP de notre contrôleur SDN. Le contrôleur est alors fonctionnel et nous pouvons continuer la configuration du point d'accès Wi-Fi OpenFlow virtuel. L'étape cruciale est la connexion du contrôleur SDN, au point d'accès Wi-Fi OpenFlow, cette communication s'établit sur le protocole de transport TCP sur le port 6633 (port officiel IANA). Sur le point d'accès Wi-Fi OpenFlow virtuel, nous lançons la commande :

```
ovs-vsctl set-controller brX tcp:10.0.0.80:6633
```

On revérifie le statut du bridge OVS à travers la commande *ovs-vsctl show*:

```
1ab8ae7e-e9da-4af9-92f6-03b5245d2544
    Bridge brX
Controller "tcp:10.0.0.80:6633"
is_connected: true
        Port "eth0"
            Interface "eth0"
        Port "eth1"
            Interface "eth1"
        Port brX
            Interface brX
                type: internal
```

Open vSwitch nous indique que le point d'accès Wi-Fi OpenFlow virtuel ainsi que le contrôleur SDN ONOS sont connectés. Nous choisissons pour la validation de cette architecture de créer deux points d'accès Wi-Fi OpenFlow virtuels distincts sur deux MNetBox différentes toutes deux reliées au contrôleur.

3.4.5 L'architecture proposée

Dans cette architecture [41], nous proposons l'instanciation de deux points d'accès Wi-Fi OpenFlow virtualisés sur deux substrats physiques permettant la virtualisation d'AP Wi-Fi que nous avons développé dans le Chapitre 1. Comme l'illustre la Figure 31, les points d'accès Wi-Fi OpenFlow virtuels se trouvent sur le plan de données en assurant le forwarding aux clients. Le contrôleur, quant à lui, se trouve sur le plan de contrôle et assure la vision globale de la topologie, des AP et des clients.

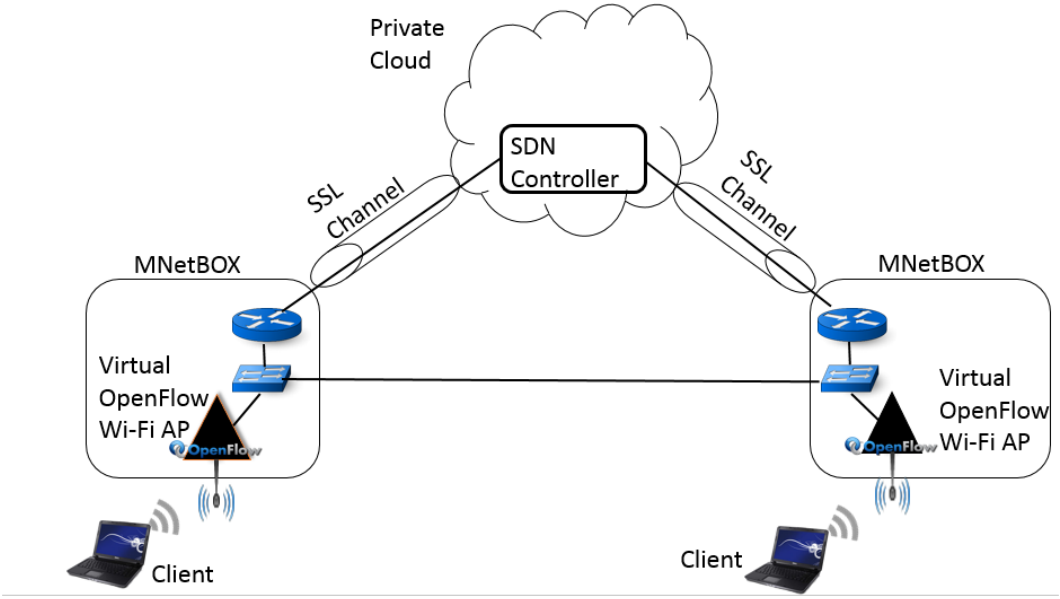

Figure 31 : Architecture des points d'accès Wi-Fi OpenFlow virtuels

Après avoir mis en place le contrôleur et les points d'accès comme décrit dans les sections précédentes, on « allume » les points d'accès Wi-Fi en lançant le daemon hostapd. Dans cette configuration, le hostapd ne nécessite aucune modification. Nous nous sommes contentés de créer un simple AP Wi-Fi protégé par une clé secrète (WPA-PSK). Une fois les interfaces Wi-Fi virtuelles fonctionnelles, l'infrastructure est prête à accueillir des clients mobiles. Leurs trafics seront ainsi gérés par le contrôleur ONOS grâce aux interactions SDN

complexes mises en place dans cet environnement virtualisé comme on peut le voir dans la Figure 32.

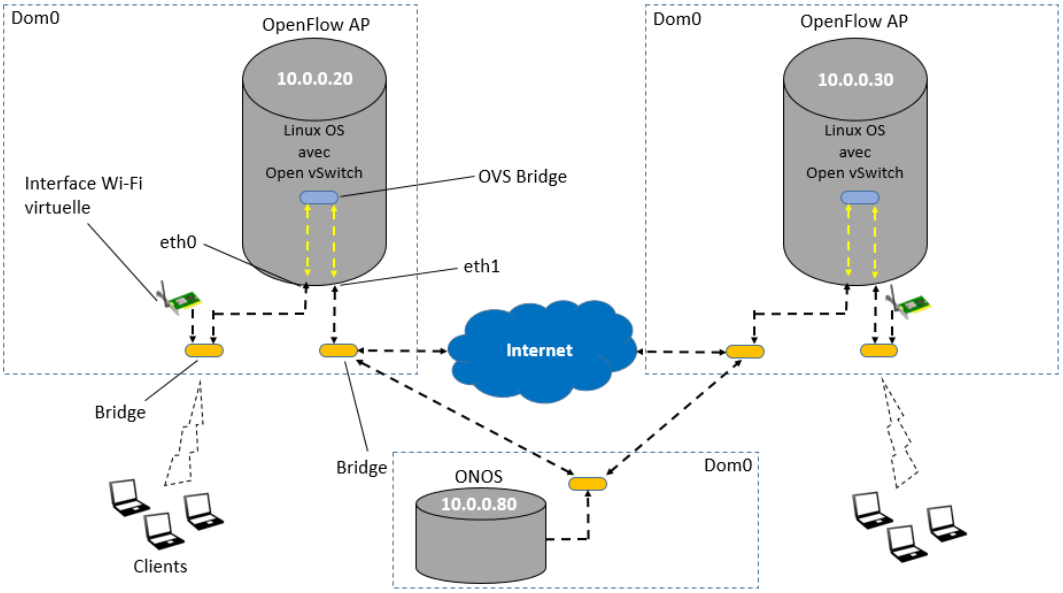

Figure 32 : Topologie réseau de 2 points d'accès Wi-Fi OpenFlow virtuels avec le contrôleur ONOS

ONOS dispose d'une interface web permettant de visualiser l'état du réseau, les points d'accès Wi-Fi fonctionnels et de contrôler les flux circulants. Figure 33 et Figure 34.

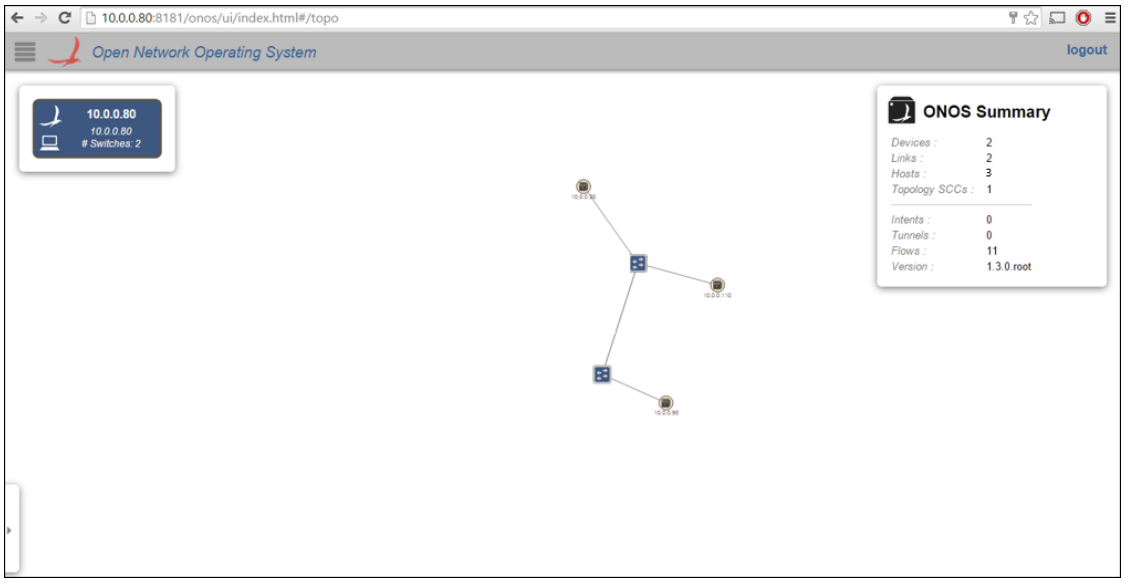

Figure 33 : Interface web ONOS pour visualiser la topologie du réseau

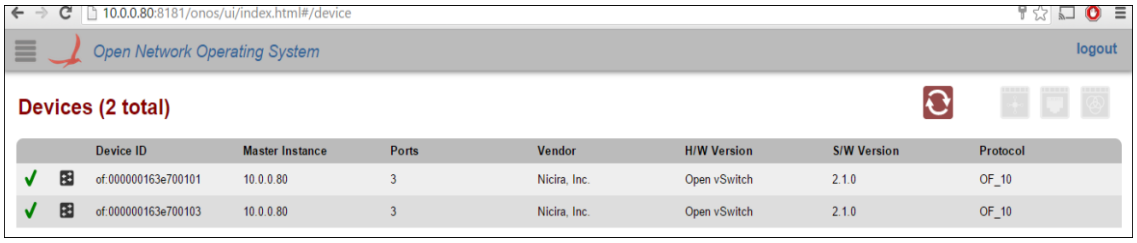

Figure 34 : Interface web ONOS pour visualiser les AP connectés

Nous décidons de connecter 3 machines clients aux points d'accès Wi-Fi OpenFlow : 2 clients sur un même point d'accès Wi-Fi et le troisième sur l'autre point d'accès, qu'on nommera respectivement A, B et C. Nous testons les pings suivants A←→B, A←→C et B←→C, qui sont tous les trois concluants.

Accès à internet

L'accès à internet est aussi possible grâce à l'application *onos-app-reactive-routing* qui est en charge du traitement et de l'installation du chemin de routage quand ONOS reçoit des IPv4 ou IPv6 packet-in. Cette application de routage réactif prend en charge les trois cas suivants :

- Un client veut envoyer du trafic vers un autre client, les deux clients se trouvent sur le réseau SDN.
- Un client dans le réseau SDN veut envoyer du trafic à une machine sur internet.
- Une machine sur internet veut envoyer du trafic à un client sur le réseau SDN.

Les chemins de routage pour le trafic de transit (le trafic d'un pair BGP en dehors du réseau SDN traverse le réseau local SDN et va à un autre pair BGP) sont proactivement installés par l'application SDN-IP (l'application *onos-app-sdnip* est aussi nécessaire).

Dans les réseaux IP traditionnels, les hôtes utilisent la passerelle comme routeur par défaut pour accéder à Internet. Cependant, un réseau SDN utilise des commutateurs SDN pour se connecter à un réseau au lieu des routeurs. Il n'y a donc pas de passerelle physique dans le réseau SDN. Sans passerelle, un problème

se pose pour les machines à l'intérieur du réseau SDN. Lorsque les hôtes veulent communiquer avec d'autres hôtes dans différents sous-réseaux, ils ne connaissent pas le prochain saut où les paquets doivent être envoyés. Les hôtes ne connaissent pas non plus l'adresse MAC du prochain saut et ne peuvent pas composer l'ensemble du paquet à envoyer. Pour résoudre ce problème, une passerelle virtuelle pour les réseaux SDN a été conçue par ONOS.

Après qu'un hôte obtient l'adresse de la passerelle, il envoie un paquet ARP pour trouver l'adresse MAC. Comme il n'y a pas de passerelle physique dans SDN, le module virtuel de la passerelle dans ONOS prendra soin de toutes les requêtes ARP. En réalité, le module de la passerelle virtuelle d'ONOS enregistre tous les ARP packet-ins. Il va vérifier si l'adresse cible dans le paquet de la requête ARP est l'adresse de la passerelle virtuelle. Si oui, le module passerelle virtuelle compose le paquet de réponse ARP et de l'envoie comme packet-out à l'hôte. Cette configuration doit se faire avec les préfixes des IP de chaque point d'accès Wi-Fi virtuel dans le fichier *sdnip.json* sous l'arborescence *onos/tools/package/config/sdnip.json.*

3.4.6 Les avantages fonctionnels de la solution

Nous avons à travers les fonctions de NFV, virtualisé l'ensemble de l'architecture SDN. Le point d'accès Wi-Fi OpenFlow et le contrôleur ONOS sont déjà prêt à l'emploi dans des machines virtuelles. Ce qui veut dire qu'après tout le travail de développement et de configuration que nous avons établis, ces VMs peuvent être dupliquées et déployées à grande échelle (moyennant un changement des adresses MACs, IPs et UUIDs des OVS pour éviter les conflits). C'est cet aspect là, qui rend notre solution originale et innovante. N'importe quel utilisateur, expérimenté ou non, peut instancier à la demande un réseau Wi-Fi OpenFlow virtuel. Nous pouvons résumer les avantages de cette solution en ces quelques points :

- Il n'est pas nécessaire d'avoir des connaissances poussées en réseau ou en système informatique pour pourvoir mettre en place

des topologies SDN Wi-Fi complexes (car nous venons de faire tout le travail en amont).

- Le déploiement d'une architecture SDN Wi-Fi devient simple et rapide.
- Il est inutile d'acheter du matériel dédié pour pouvoir créer un SDN Wi-Fi. Un substrat physique supportant la virtualisation des points d'accès Wi-Fi suffit
- La flexibilité est accrue, cette solution permet de créer et de supprimer des réseaux SDN Wi-Fi.

3.5 Conclusion

La promesse faite par SDN est de rendre les réseaux ouverts et programmables. Mais l'ampleur de cette ouverture et de cette flexibilité dépend en fin de compte de la mise en pratique des vendeurs et de leurs respects des normes. Les implémentations limitées ou les extensions propriétaires serviront seulement à paralyser le progrès des réseaux SDN. Pour tenir sa promesse, SDN doit être pensé pour tous les utilisateurs et à travers tous les types de réseaux, avec une véritable interopérabilité entre les composants du réseau via OpenFlow. Avec un accès programmable à l'infrastructure Wi-Fi, des applications de réseau pourront désormais communiquer directement avec les contrôleurs Wi-Fi jusque lors point d'ombre des réseaux. En outre, l'intégration des fonctionnalités de CAPWAP dans OpenFlow pourrait ouvrir de nouveaux horizons pour la qualité d'expérience des clients mobiles grâce à une itinérance maitrisée. La solution que nous avons développée permet l'utilisation des points d'accès Wi-Fi OpenFlow virtuels pour accéder aux mêmes fonctionnalités (accès à internet) qu'un point d'accès Wi-Fi traditionnel. La présence du contrôleur SDN, ayant une vue globale sur la topologie, les clients, les AP, les flux entrants et sortants, offre un grand tremplin pour l'innovation des réseaux Wi-Fi à travers les avancées des réseaux SDN.

Conclusion générale

Résumé des contributions

La virtualisation originalement introduite dans les réseaux pour en réduire les coûts de maintenance et de déploiement, a connu une explosion fulgurante remodelant le paysage des réseaux informatiques et télécoms. La virtualisation s'est imposée comme une alternative pour bâtir les nouvelles solutions informatique du futur. Les progrès des technologies de virtualisation ont créé des nouvelles possibilités pour les opérateurs de télécommunication pour tirer parti des ressources physique de manière plus efficace. Plusieurs solutions de virtualisation basées sur des technologies hétérogènes voient le jour. L'autre constat de taille concerne la mobilité du client qui devient de plus en plus mobile et nécessite un accès à ses services en tout lieu et tout le temps. Le réseau ainsi se métamorphose pour permettre aux clients de se connecter à travers ses différents équipements. Le Wi-Fi quant à lui est déployé de façon ubiquitaire, et est devenu le moyen d'accès à Internet le plus utilisé. Cette thèse a exploré ces deux tendances globales pour proposer des points d'accès Wi-Fi virtuels reflétant la métamorphose du réseau. L'urbanisation des réseaux a été notre fil conducteur tout au long de ce manuscrit, pour proposer d'avantages de fonctionnalités dans les points d'accès virtuels permettant une mutualisation de l'infrastructure matérielle. Dans le Chapitre 1, nous avons présenté la virtualisation et l'hyperviseur Xen dans leurs généralités. Ceci nous a donné des bases solides pour comprendre par la suite le concept de points d'accès Wi-Fi virtuels développés et tout l'enjeu qu'ils constituent dans l'urbanisation des réseaux d'accès.

Nous avons vu par la suite dans le Chapitre 2 que les réseaux Wi-Fi représentaient un réel intérêt pour les opérateurs pour consolider leurs réseaux en délestant les données des bandes 3G/4G sur les bandes Wi-Fi. Le problème constaté est que l'accès au Wi-Fi public dans les Hotspots est non sécurisé, contraignant à configurer à travers les portails captifs et non standardisé. L'émergence de la norme Hotspot2.0 a été prévue à cet effet. Notre contribution a consisté en la virtualisation de cette nouvelle technologie sur notre environnement

de virtualisation. Nous avons rapidement vu les limites de cette nouvelle technologie, apportant un niveau de sécurité d'entreprise dans des lieux publics. Comment fournir les outils d'authentification adéquats aux clients dans un réseau Wi-Fi public ? L'authentification EAP-TLS obligeant le client à avoir des certificats électroniques, nous a mené à la création d'une architecture à base de bornes NFC. Les bornes NFC permettent -après vérification de l'identité du client- de créer et rapatrier tous les droits d'accès sans aucune intervention manuelle. Dans la même philosophie, une autre proposition faite par cette thèse est une architecture à base de bornes NFC permettant de créer à la volée des points d'accès Wi-Fi virtuels dans des boxs publiques sans aucune manipulation de la part du client. Cette dernière architecture reflète parfaitement la métamorphose des architectures de réseau pour permettre l'urbanisation des points d'accès Wi-Fi virtuels (à travers la mutualisation d'un matériel opérateur).

Une autre problématique abordée par cette thèse est la capacité du réseau d'accès virtualisé à suivre les innovations que connaissent les réseaux cœurs virtualisés à travers SDN. La configuration, la mise en place et le déploiement de points d'accès Wi-Fi à grande échelle (réseaux d'entreprise et de campus) sont des tâches fastidieuses sujettes aux erreurs introduites par la manipulation intensive et manuelle. Nous avons vu aussi à travers une description des architectures Wi-Fi traditionnelles que l'interopérabilité entre les AP proposés par différents fournisseurs n'était pas du tout respectée à cause des extensions propriétaires ajoutées au-dessus de standards normalisés. Nous avons à travers une dernière contribution virtualisé l'architecture SDN dans nos équipements, pour permettre de franchir ces problèmes. Ainsi les points d'accès Wi-Fi OpenFlow, et le contrôleur SDN ONOS virtuel que nous avons développés permettent de bâtir une solution évolutive pour les réseaux Wi-Fi virtuels poussé par le potentiel prometteur de SDN. SDN sur les réseaux d'accès permet une auto-configuration et une flexibilité qui manquait jusqu'alors sur la bordure.

Perspectives

Ce travail ouvre la voie à plusieurs perspectives à la fois de recherche et industrielles qui mériteraient d'être explorées.

Le développement d'un prototype de borne NFC permettant le rapatriement des éléments d'authentification pour se connecter à un point d'accès Wi-Fi hautement sécurisé serait bénéfique à l'entreprise VirtuOR. En effet cette architecture permet de donner une solution complète d'urbanisation de point d'accès, allant de la création du point d'accès, au provisionnement des droits d'accès, à travers un échange NFC.

Aujourd'hui la norme Hotspot2.0 ne connait plus le même engouement de la part des industriels et des opérateurs qu'à sa création. L'adoption du Hotspot2.0 est assez complexe techniquement comme nous l'avons vu, mais surtout elle suppose une mutualisation de certaines infrastructures qui rend récalcitrants la plupart des industriels. La monétisation d'une telle solution est aussi problématique, et n'a pas encore était abordée dans la littérature. Cependant il serait très judicieux de continuer la recherche sur une telle technologie qui innove réellement l'expérience Wi-Fi, en attendant que les industries l'adoptent.

Et enfin l'une des ouvertures les plus prometteuses faite par cette thèse est le Wi-Fi SDN est toute l'évolutivité qu'elle constitue. Nous avons pu réaliser une plateforme Wi-Fi SDN virtuelle fonctionnelle lors de nos travaux. Ce que nous cherchions à faire avant tout est de prouver la validité d'une telle solution, et d'en démontrer l'usage. Des nouvelles versions des protocoles SDN orientées Wi-Fi permettront dans un avenir proche de dynamiser et remodeler les réseaux Wi-Fi, qui sont devenus indéniablement partie intégrante du paradigme de réseau SDN.

Liste des publications :

- **Démo**
  - O. Stiti and O. Braham, "Demonstration: Virtual access network to the cloud". *12th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*. June 2013.
- **Journal**
  - O. Stiti, O. Braham and G. Pujolle. « Creation of Virtual Wi-Fi Access Point and Secured Wi-Fi Pairing, through NFC", *International Journal of Communications, Network and System Sciences*, Vol.7 No.6 175-180, June 2014.
- **Conférences et workshops**
  - O. Stiti and O. Braham, "Initialization of Wi-Fi CERTIFIED Passpoint devices based on EAP-TLS with NFC terminals", *First Workshop of Mobile applications, SeCure Elements and Near Field Communication*. February 2014.
  - O. Stiti, O. Braham and G. Pujolle, "On-Demand instantiation of an OpenFlow-Based SDN in Wi-Fi Network", *International workshop Software Networks (SoftNet),* March 2015.
  - O. Stiti, O. Braham and G. Pujolle, "Virtual OpenFlow-based SDN Wi-Fi Access Point", *Global Information Infrastructure and Networking Symposium (GIIS)*, October 2015.

Liste des figures

Liste des tables

Références

[1] 1E, “Server energy and efficiency report”, *Alliance to save the energy*, 2009

[2] C. Gabriel, “Wireless broadband alliance industry report 2013: Global trends in public wi-fi”, *Marvedis Rethink-Wireless Broadband Alliance*, 2013.

[3] Cisco, “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update 2014–2019”, *White Paper*, 2015.

[4] Cisco, “Evolution of the Mobile Network”, 2010.

[5] Wi-Fi Alliance, “Hotspot 2.0”, Release 1, *Technical Specification*, Version 1.0.0, 2012

[6] O. Braham and G. Pujolle, “The metamorphosing network (M-Net)”, *Global Information Infrastructure and Networking Symposium (GIIS)*, 2012.

[7] ETSI, “Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges & Call for Action”, White paper, *SDN and OpenFlow World Congress*, October 2012.

[8] P.V.V. Reddy and L. Rajamani, "Virtualization overhead findings of four hypervisors in the CloudStack with SIGAR," in *Information and Communication Technologies (WICT),* December *2014.*

[9] O. Stiti and O. Braham, “Demonstration: Virtual access network to the cloud”. *12th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*. 2013.

[10] Tout sur la faille de sécurité Heartbleed, http://www.heartbleed.fr/

[11] Wi-Fi Alliance. “Wi-Fi Simple Configuration Protocol and Usability Best Practices for the Wi-Fi Protected Setup™ Program”, Version 2.0.1, Avril 2011.

[12] S. Clark, “One in three mobile phones to come with NFC by 2017”. *NFC World*, June 2013.

[13] Atmel Corporation, “Requirements of ISO/IEC 14443 Type B Proximity Contactless Identification Cards”, 2005.

[14] N. Asokan, V. Niemi, and K. Nyberg, “Man-in-the-middle in tunnelled authentication protocols”. Springer-*Security Protocols*, pp. 28-41, 2005

[15] P. Funk, and S. Blake-Wilson, "Eap tunneled tls authentication protocol (EAP-TTLS)." *Work in Progress*, 2004.

[16] P. Pourghomi and G. Ghinea. “Managing NFC payment applications through cloud computing”, *IEEE International Conference for Internet Technology and Secured Transactions (ICITST)*, December 2012.

[17] M. Pasquet, J. Reynaud and C. Rosenberger. "Secure payment with NFC mobile phone in the SmartTouch project", *International Symposium on Collaborative Technologies and Systems (CTS),* May 2008.

[18]: S. Viehböck. "Brute forcing Wi-Fi Protected Setup, when poor design meets poor implementation". *White paper*, December 2011.

[19] P. Hoffman, "SMTP Service Extension for Secure SMTP over Transport Layer Security (SMTP-TLS)", *IETF, RFC 3207*, February 2002.

[20] C. Newman, "Using TLS with IMAP, POP3 and ACAP", *IETF*, *RFC 2595*, June 1999.

[21] E. Rescorla. "HTTP Over TLS", *IETF, RFC 2818,* May 2000.

[22] Security Risks of Near Field Communication, http://www.nearfieldcommunication.org/nfc-security-risks.html

[23] O. Stiti and O. Braham, "Initialization of Wi-Fi CERTIFIED Passpoint devices based on EAP-TLS with NFC terminals", *First Workshop of Mobile applications, SeCure Elements and Near Field Communication.* February 2014.

[24] O. Stiti, O. Braham and G. Pujolle. « Creation of Virtual Wi-Fi Access Point and Secured Wi-Fi Pairing, through NFC", *International Journal of Communications, Network and System Sciences,* Vol.7 No.6 175-180, June 2014.

[25] O. M. E. Committee, "Software-defined networking: The new norm for networks." *Technical report, Open Networking Foundation*, 2012.

[26] H. Kim and N. Feamster, "Improving network management with software defined networking", *IEEE Communications Magazine*, 2013.

[27] P. Neira-Ayuso, R. Gasca, L. Lefevre, "Communicating between the kernel and user-space in Linux using Netlink sockets", *Software: Practice and Experience*, *40*(9), 797-810. 2010

[28] Meru Networks. "SDN for Wi-Fi OpenFlow-enabling the wireless LAN can bring new levels of agility", *White paper*, 2014.

[29] M. Mendonca, K. Obraczka, and T. Turletti, "The case for software-defined networking in heterogeneous networked environments", *Proceedings of the 2012 ACM conference on CoNEXT student workshop* (pp. 59-60), December 2012.

[30] P. Calhoun, M. Montemurro and D. Stanley, "Control And Provisioning of Wireless Access Points (CAPWAP) Protocol Specification", *IETF*, *RFC 5415,* March 2009.

[31] C. Rotsos, N. Sarrar, S. Uhlig, R. Sherwood, and A. W. Moore, "OFLOPS: An open framework for OpenFlow switch evaluation", *Springer Berlin Heidelberg , Passive and Active Measurement,* pp. 85-95, 2012.

[32] Open Network Fundation, "Openflow switch specification version 1.3.3", *Technical report,* 2013.

[33] Cisco, “Troubleshoot Common Problems with Wireless Bridged Networks”, *Document ID: 22950*, 2009.

[34] K. Lin. “Interoperability Considerations for Today’s Wi-Fi Networks”, *White Paper*, 2007.

[35] J. Schulz-Zander, N. Sarrar and S. Schmid, “Aeroflux: A near-sighted controller architecture for software-defined wireless networks”. *Proc. Open Networking Summit (ONS),* 2014.

[36] S. Monin, A. Shalimov and R. Smeliansky, “Chandelle: Smooth and fast WiFi roaming with SDN/OpenFlow”, *US Ignite*, 2014.

[37] Production Quality, Multilayer Open Virtual Switch, http://openvswitch.org/

[38] J. Duffy, “OpenDaylight: Where's the love?”, *http://www.networkworld.com/article/2174104/lan-wan/opendaylight--where-s-the-love-.html.* , February 2014

[39]: J. Duffy, “ON.Lab making its own open source SDN operating system available”, *http://www.networkworld.com/article/2842859/sdn/atandt-others-launch-opendaylight-sdn-alternative.html.,* November 2014.

[40] Products, www.virtuor.fr

[41] O. Stiti, O. Braham and G. Pujolle, « Virtual OpenFlow-based SDN Wi-Fi Access Point”, *Global Information Infrastructure and Networking Symposium (GIIS)*, October 2015.

[42] P. Urien, “LLCPS”, *IETF draft*, 2012.

[43] NFC Forum Specifications*, http://www.nfc-forum.org/specs/*

[44] P. Urien, “LLCPS: A New Security Framework Based On TLS For NFC P2P Applications In The Internet Of Things”, *IEEE CCNC*, 2013.

[45] J. Malinen, “hostapd: IEEE 802.11 AP, IEEE 802.1 X. WPA/WPA2/EAP/RADIUS Authenticator”, *http://hostap.epitest.fi/hostapd*, 2014.

[46] M. Vipin and S. Srikanth, “Analysis of open source drivers for IEEE 802.11 WLANs”, *IEEE International Conference on Wireless Communication and Sensor Computing*, ICWCSC, pp. 1-5, January 2010.

[47] C. Avin, M. Borokhovich, and A. Goldfeld, “Mastering (Virtual) Networks” *Proc. of CSEDU* , p. 250-257, 2009.

[48] S. M. Lee, S. B. Suh, B. Jeong, S. Mo, B. M. Jung, J. H. Yoo and D. H. Lee, “Fine-grained i/o access control of the mobile devices based on the xen architecture” *ACM 15th annual international conference on Mobile computing and networking*, pp. 273-284, September 2009.

[49] M. Vipin and S. Srikanth, “Analysis of open source drivers for IEEE 802.11 WLANs” *IEEE International Conference on Wireless Communication and Sensor Computing (ICWCSC),* pp. 1-5, January 2010.

[50] Quagga Routing Suite, *http://www.quagga.net.*

[51] T. L. Swan and D. U. McKinney, "Providing quality of service (QOS) using multiple service set identifiers (SSID) simultaneously", *U.S. Patent No. 8,184,530*, 22 May 2012.

[52] WG802.11 - Wireless LAN Working Group, "802.11u-2011 - IEEE Standard for Information Technology-Telecommunications and information exchange between systems-Local and Metropolitan networks-specific requirements-Part II: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 9: Interworking with External Networks", *IEEE Standard*, 2011.

[53] WG802.11 - Wireless LAN Working Group, "802.11i-2004 - IEEE Standard for information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements-Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment 6: Medium Access Control (MAC) Security Enhancements", *IEEE Standard*, 2004.

[54] D. Simon, B. Aboba and R. Hurst, "The EAP-TLS Authentication Protocol", *IETF, RFC 5216*, March 2008.

[55] H. Haverinen and J. Salowey, "Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM)", *IETF, RFC 4186*, January 2006.

[56] J. Arkko and H. Haverinen, "Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA)" *IETF, RFC 4187*, January 2006.

[57] P. Funk and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", *IETF, RFC 5281*, August 2008.

[58] G. Zorn, "Microsoft PPP CHAP Extensions, Version 2", *IETF, RFC 2759*, January 2000.

[59] C. Rigney, S. Willens, A. Rubens and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", *IETF, RFC 2865*, June 2000.

[60] D. Whiting, R. Housley and N. Ferguson, "Counter with CBC-MAC (CCM)", *IETF, RFC 3610*, September 2003.

[61] S. Armitage and A. Buxey, "Improving the eduroam experience with 802.11u", *TERENA Networking Conference (TNC),* 2013.

[62] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide and G. Parulkar, "ONOS: towards an open, distributed SDN OS" *ACM third workshop on Hot topics in software defined networking*, pp. 1-6, August 2014.